International Telecommunication Union

ITU-T G.7701

(04/2022)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

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

Data over Transport – Generic aspects – Transport network control aspects

Common control aspects

Recommendation ITU-T G.7701

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

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

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

Recommendation ITU-T G.7701

Common control aspects

Summary

Recommendation ITU-T G.7701 describes the concepts and the aspects of management control components that are common to the use of either software defined networking (SDN) and automatically switched optical network (ASON) approaches to the management of a transport network. It also describes the common aspects of the interaction between the management-control functions and the transport network resources.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.7701	2016-11-13	15	11.1002/1000/13090
1.1	ITU-T G.7701 (2016) Amd. 1	2018-03-16	15	11.1002/1000/13539
1.2	ITU-T G.7701 (2016) Amd. 2	2020-12-22	15	11.1002/1000/14524
2.0	ITU-T G.7701	2022-04-06	15	11.1002/1000/14928

Keywords

ASON, MC components, management-control continuum, transport SDN.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, 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 appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T G.7701

Common control aspects

1 Scope

This Recommendation describes the concepts and the aspects of management and control (MC) components that are common to both software defined networking (SDN) [ITU-T G.7702] and automatically switched optical network (ASON) [ITU-T G.7703] to control transport networks. This encompasses common aspects of:

– transport resources and their representation;

– MC components used to describe architecture for application of SDN and ASON to the control of transport networks;

– control communication aspects;

– common management aspects of MC components;

– identifiers with respect to naming and addressing, which includes the separation of identity from location.

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.698.1] Recommendation ITU-T G.698.1 (2009), *Multichannel DWDM applications with single-channel optical interfaces.*

[ITU-T G.698.2] Recommendation ITU-T G.698.2 (2018), *Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces*.

[ITU-T G.800] Recommendation ITU-T G.800 (2016), *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.872] Recommendation ITU-T G.872 (2019), *Architecture of optical transport networks*.

[ITU-T G.7702] Recommendation ITU-T G.7702 (2022), *Architecture for SDN control of transport networks.*

[ITU-T G.7703] Recommendation ITU-T G.7703 (2021), *Architecture for the automatically switched optical network.*

[ITU-T G.7710] Recommendation ITU-T G.7710 (2020), *Common equipment management function requirements.*

[ITU-T G.7711] Recommendation ITU-T G.7711/Y.1702 (2022), *Generic protocol-neutral information model for transport resources*.

[ITU-T G.7712] Recommendation ITU-T G.7712/Y.1703 (2019), *Architecture and specification of data communication network.*

[ITU-T G.7714.1] Recommendation ITU-T G.7714.1/Y.1705.1 (2017), *Protocol for automatic discovery in transport networks*.

[ITU-T G.7716] Recommendation ITU-T G.7716 (2010), *Architecture of control plane operations*.

[ITU-T G.7718] Recommendation ITU-T G.7718 (2020), *Framework for the management of management-control components and function.*

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

[ITU-T Q.2982] Recommendation ITU-T Q.2982 (1999), *Broadband integrated services digital network (B-ISDN) – Digital subscriber signalling system No. 2 (DSS2) – Q.2931-based separated call control protocol*.

[ITU-T X.509] Recommendation ITU-T X.509 (2019), *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[ITU-T X.731] Recommendation ITU-T X.731 (1992), *Information technology – Open Systems Interconnection – Systems management: State management function.*

[ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 access point: [ITU-T G.800].

3.1.2 adaptation: [ITU-T G.800].

3.1.3 characteristic information (CI): [ITU-T G.800].

3.1.4 connection: [ITU-T G.800].

3.1.5 control communication network: [ITU-T G.7712].

3.1.6 data communication network: [ITU-T G.7712].

3.1.7 forwarding point (FP): [ITU-T G.800].

3.1.8 layer network: [ITU-T G.805].

3.1.9 link: [ITU-T G.805].

3.1.10 link connection: [ITU-T G.805].

3.1.11 logical termination point: [ITU-T G.7711].

3.1.12 software defined networking: [ITU-T Y.3300].

3.1.13 subnetwork: [ITU-T G.805].

3.1.14 subnetwork connection: [ITU-T G.805].

3.1.15 trail: [ITU-T G.805].

3.1.16 transitional link: [ITU-T G.800].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 address: A string of symbols that is valid regardless of the location of the source but changes if the destination moves. An address is used for the purpose of routing. Source and destination addresses must be globally unique.

3.2.2 administrative domain: A type of domain that belongs to a single entity such as a network operator, a service provider or an end-user.

3.2.3 allocated (resource) label range: 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 resource label range. The allocated labels are entities that can be referenced in the transport resource name space. Each allocated label is associated with one or multiple subnetwork point (SNP) IDs that exist in the MC component name space (1:n relationship). Termination and adaptation performers (TAPs) hold this binding information.

3.2.4 automatically switched optical network (ASON): A connection-oriented circuit switched (CO-CS) or connection-oriented packet switched (CO-PS) transport network, as defined in [ITU-T G.800], together with a set of distributed management-control (MC) components, that cooperate to configure the transport network.

3.2.5 boundary resource identifier (BRI): Used at a domain boundary to identify both the transport resources, and the interface(s) between MC components, at that boundary.

3.2.6 call: An association between two or more users and one or more domains that support an instance of a service through one or more domains. Within domains, the association is supported by network entities that contain call state. Between a user and a network call control entity and between network call control entities, there are call segments. The call consists of a set of concatenated call segments.

3.2.7 call admission control: A policy function invoked by an originating role in a network and may involve cooperation with the terminating role in the network.

3.2.8 call control: An association between one or more user applications and the network to control the set-up, release, modification and maintenance of sets of connections.

3.2.9 call controller: An MC component that manages the setup, release, modification and maintenance of calls across a transport network. There are two types of call controller, the calling/called party call controller (CCC) and the network call controller (NCC).

3.2.10 call segment: An association between two call control entities (as per [ITU-T Q.2982]). Each call segment has zero or more associated connections. Call segments between network call control components have zero or more supporting calls.

3.2.11 calling/called party call controller: A type of call controller that is associated with an end system that initiates or terminates a call. It may be co-located with an end system or located remotely and act as a proxy on behalf of an end system. A calling/called party call controller has two roles, one to support the calling party and the other to support the called party.

3.2.12 configured (resource) label: A transport resource label that has been configured in support of a connection. For each configured label, a forwarding table entry exists on the receiving end of the link such that packets can be forwarded based on the label value of the received packet.

3.2.13 connection admission control: A process that determines if there are sufficient resources to admit a connection (or re-negotiates resources during a call).

3.2.14 connection controller (CC): The MC component responsible for coordination among the link resource manager, the routing controller and both peer and subordinate connection controller components for the purpose of the management and supervision of connection setups, releases and the modification of connection parameters for existing connections.

3.2.15 control domain: A type of domain that represents the scope of control over a collection of resources allocated to that domain as determined by a set of functions from the management-control continuum.

3.2.16 discovery agent (DA): The "federation" of discovery agents operates in the transport resource name space and provides for separation between that space and the MC component name space. The federation has knowledge of forwarding points (FPs) in the network, while a local DA has knowledge of only those FPs assigned to it.

3.2.17 domain: Represents a collection of resources that are grouped for a particular purpose. Domains of the same type do not overlap among themselves, but a hierarchical containment relationship of domains is possible.

NOTE – In general, resources may be considered to encompass compute, networking, and storage.

3.2.18 link resource manager (LRM): An MC component responsible for the management of a subnetwork point pool (SNPP) link, including the allocation and un-allocation of subnetwork point (SNP) link connections, managing resource reservation, the configuration of policing and shaping functions via the termination and adaptation performer (TAP) (if required), providing topology and status information.

3.2.19 MC component: An abstract representation of a functional entity. In this Recommendation, MC components do not represent instances of implementation code. They are used to construct scenarios to explain the operation of the architecture.

3.2.20 MC component interface: Interfaces that represent the logical relationships between MC components and are defined by the information flow between these entities.

NOTE – Such a relationship allows for choice in selection of potentially exposable interfaces in support of different implementations and network architectures.

3.2.21 network call controller (NCC): A type of call controller that exists at domain boundaries. It supports three roles, one to support of the calling party, another to support the called party and a third to support calls across domain boundaries.

3.2.22 node: In the context of this Recommendation, the term "node" is used to signify a subnetwork or a routing area.

3.2.23 notification component: An MC component, which processes notifications from other MC components in the same MC system and forwards them to one or more external clients.

3.2.24 policy: The set of one or more rules that define the action that an MC component takes in response to a set of conditions presented at an interface. More than one policy may be applied (sequentially) to these conditions to determine the final action taken by the MC component.

3.2.25 potential (resource) label range: Full label range of resource labels in the transport resource name space that an adaptation function supports to distinguish different information flows.

3.2.26 potential subnetwork points (SNPs): Those SNPs that are associated with a (resource) label.

3.2.27 protocol controller (PC): An MC component that maps the parameters from one or more interfaces of one or more MC components into messages that are carried by a protocol across the CCN. It also demultiplexes messages received from the CCN into parameters that are passed to other MC components. The protocol controller is attached to the CCN via a reference point (for example a CPI, UNI, I-NNI, E-NNI).

3.2.28 provisioning: The act of specifying the parameters necessary when assigning/deassigning network resources to/from the management and control (MC) component or to invoke/remove services provided by an MC component instance. These parameters are specific to a resource or service request, causing changes to these parameters to only impact a specific resource or service request. Therefore, provisioning is allowed in the initialization and operations phases of the MC component lifecycle.

3.2.29 recovery domain: A type of control domain whose purpose is assuring the reliable transfer of information across the resources allocated to that domain.

NOTE – Approaches to support reliable transfer of information may include, e.g., protection, distributed or centralized restoration, etc. A pre-condition for establishing a recovery domain is that there are sufficient network resources in the domain to maintain a specified level of reliability for the selected approach(es). In multi-layer transport networks, a recovery domain in a client layer network must fully contain any recovery domains in the server layer networks that it uses.

3.2.30 resource database (RDB): A logical entity that holds (and makes available) information that MC components use for their operations. Related MC components share the same RDB.

3.2.31 route: A sequence of the artefacts representing the transport resources (i.e., SNPs, SNPPs, and routing areas) that support, or are intended to support, a connection.

3.2.32 routing: A control function used to select paths for the establishment of connections through one or more operator networks.

3.2.33 routing area (RA): An instance of a routing domain, and 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 a subnetwork.

3.2.34 routing controller (RC): An MC component with the roles 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 management-control continuum purposes.

3.2.35 routing domain: A type of control domain whose purpose is routing across the collection of resources allocated to that domain.

3.2.36 routing level: A relationship between a routing area (RA) and a containing RA or contained RAs. The containment hierarchy of routing areas creates routing levels.

3.2.37 service level agreement: A contract between two parties such as a service provider and a customer. It defines the services available to the customer, and the grade of service of those services as offered to the customer. It also usually describes the service guarantee and potential penalties in case of service degradation or failure.

3.2.38 subnetwork point (SNP): A control abstraction that represents an actual or potential underlying forwarding point (FP) or forwarding end point (FwEP) (see [ITU-T G.800]). Several SNPs (in different VNs) may represent the same FP or FwEP.

3.2.39 Subnetwork point (SNP) identifier: An instance of an identifier for an SNP.

NOTE – A subnetwork point (SNP) identifier is used for link connection assignment and, in some cases, routing. The SNP identifier is derived from the subnetwork point pool (SNPP) identifier concatenated with a locally significant SNP index. When the identifier is routable, it is an SNPP address. When it is not routable, the identifier is an SNP name.

3.2.40 subnetwork point pool (SNPP): A set of SNPs that are grouped together.

NOTE – An SNPP usually represents a link end. An SNPP may be further subdivided (sub-structured) into smaller pools.

3.2.41 subnetwork point pool (SNPP) alias: An alternate SNPP name for the same SNPP link that is generated from another SNPP name space.

NOTE – If present in a routing area, it is available to the routing controller (RC) that is associated with the routing area (RA).

3.2.42 subnetwork point pool (SNPP) identifier: An instance of an identifier for an SNPP.

NOTE – When the identifier is routable, it is an SNPP address. When it is not routable, the identifier is an SNPP name. The constituents of an SNPP identifier may include routing area (RA) identifiers, a subnetwork identifier and resource context identifiers.

3.2.43 subnetwork point pool link (SNPP link): An association between subnetwork point pools (SNPPs) on different subnetworks.

3.2.44 termination and adaptation performer (TAP): An MC component that provides the link resource manager (LRM) with a control view (in the subnetwork point (SNP) name space) of the adaptation and termination functions (in the forwarding point (FP) name space), and abstracts any hardware and technology-specific details.

3.2.45 transitional SNPP link: A subnetwork point pool (SNPP) link with an SNPP in a subnetwork in one layer network, and an SNPP in a subnetwork of a different layer network. It may also be an SNPP link with an SNPP in a subnetwork in one sublayer and an SNPP in a subnetwork of a different sublayer, where both sublayers are in the same layer network.

3.2.46 virtual network (VN): A designated subset of abstracted network resources.

NOTE – The network resources in the VN may be at different levels of abstraction and may correspondingly use identifiers from different name spaces.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASON Automatically Switched Optical Network

BRI Boundary Resource Identifier

CC Connection Controller

CCC Calling/called party Call Controller

CCN Control Communication Network

CI Characteristic Information

CIR Committed Information Rate

cFP Client Forwarding Point

CO-CS Connection-Oriented Circuit Switched

CO-PS Connection-Oriented Packet Switched

CoS Class of Service

DA Discovery Agent

DCN Data Communication Network

DM Discovery Message

EIR Excess Information Rate

EMS Element Management System

E-NNI	External Network-Network Interface
FEC	Forward Error Correction
FEF	Fault Event Filtering
FRM	Fault Reporting Management
FP	Forwarding Point
FwEP	Forwarding End Point
GFP	Generic Framing Procedure
IM	Information Model
I-NNI	Internal Network-Network Interface
LRM	Link Resource Manager
LP	Layer Protocol
LTP	Logical Termination Point
MC	Management and control
MPLS	Multiprotocol Label Switching
NCC	Network Call Controller
NE	Network Element
NMS	Network Management System
OAM	Operations, Administration and Maintenance
OCN	Overhead Communications Network
ODUk	Optical Data Unit-k
OSS	Operations Support System
OTN	Optical Transport Network
OTSi	Optical Tributary Signal
OTSiA	Optical Tributary Signal Assembly
OTU	Optical Transport Unit
OTUk	completely standardized Optical Transport Unit-k
PC	Protocol Controller
PDP	Policy Decision Point
PEP	Policy Enforcement Point
RA	Routing Area
RC	Routing Controller
RDB	Resource Database
SDN	Software Defined Networking
sFP	Server Forwarding Point
SNC	Subnetwork Connection
SNP	Subnetwork Point
SNPP	Subnetwork Point Pool

TAP	Termination and Adaptation Performer
TTP	Trail Termination Point
UML	Unified Modelling Language
UNI	User-Network Interface (reference point)
VC	Virtual Container
VCAT	Virtual Concatenation
VN	Virtual Network

5 Conventions

This Recommendation uses the diagrammatic conventions defined in [ITU-T G.800] to describe the transport network and its resources.

6 Overview

Network management (e.g., performed by an OSS), ASON control, and SDN controllers all perform the same operations on the transport resources. The distinctions among these approaches are the variations that may exist in the degree of automation (high human intervention to none), distribution of their implementation (centralized versus fully or partially distributed), and exposure of interfaces (closed versus open). However, the transport resources themselves are unable to distinguish how the connection management function is performed but simply respond to operations (commands) that they receive and report changes.

This leads to the concept whereby management and control (MC) functions are considered to be in a continuum. This is known as the management-control continuum; i.e., it expresses the view that MC functions are essentially the same and, thus, they can be grouped into one set of MC functions.

In the management-control continuum, interfaces between functions (or groups of functions) may be exposed when resources are placed in different domains, when an interface to users is desired, when a network management function operates on a multi-vendor set of resources, as well as other factors.

Figure 6-1 illustrates the management-control continuum concept and its relationship with transport resources, whereby MC functions operate on transport resources and receive state information about resources. MC function (in MC components) may be contained in MC systems of which three types are shown: SDN, ASON, and others. Within MC systems, some of the MC components are detailed in this Recommendation and are represented by the yellow triangle symbol. Each instance of an MC system has in its scope a set of transport resources over which it supports connection management in a layer network. Management functions recur and a management function instance can manage a set of other management function instances. This is depicted in the network administration role that shows the functions "Configuration of MC components and system" and "Assignment of transport resources to an MC system". The first function would be used to instantiate an MC system and its MC components, and the second function assigns transport resources to the scope of that MC system.

Transport resource management functional areas are identified in [b-ITU-T M.3010] as: performance management, fault management, configuration management, accounting management and security management. These are shown in an "Other MC Systems" in Figure 6-1 which could be an instantiated as an operations support system (OSS)/network management system (NMS)/element management system (EMS).

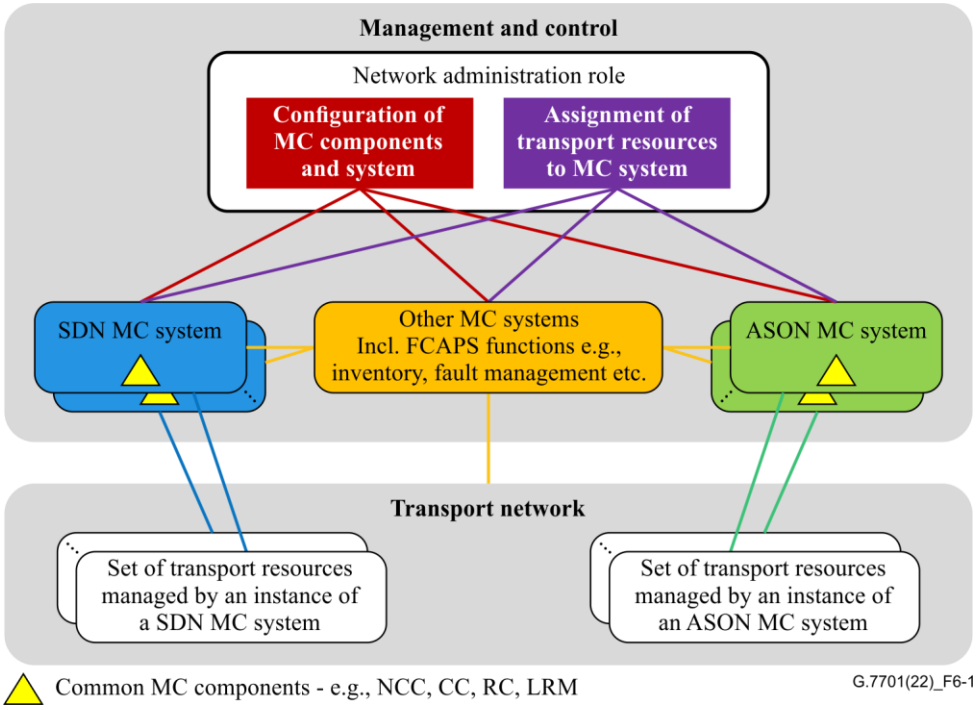


Figure 6-1 – Management-control continuum

Not shown is the data communications network (DCN), which provides the communication paths to carry, e.g., ASON control, SDN controller, and management information communications. The DCN when used in ASON and SDN architectures is referred to as the Control Communication Network (CCN) and is a separate set of transport resources also controlled by an MC system. For example, it could be an instance of an "Other MC System" which manages an IP network whose link resources are from the transport network also in Figure 6-1. The details of the CCN, management functions, and transport resources are specified in other Recommendations and are outside the scope of this Recommendation.

As shown in Figure 6-1, the transport network resources may be partitioned and assigned to an MC system, this makes those resources visible to this MC system and only this MC system should have the ability to configure those resources. Some transport network resources that are not currently in use, may be reserved for future use and may not be assigned to any MC system. The assignment of the transport network resources is performed by the network administration role and thus, is outside the scope of this Recommendation.

6.1 Call and connection control

This Recommendation separates the treatment of call control and connection control, call control is only used at domain boundaries in a single layer network or at the boundary between layer networks to support an interlayer call. Within a domain or between domains that use an I-NNI it is only necessary to support procedures for connection control. 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.

The call represents the service offered to the user, the connections (that support the call) are the means by which networks provide the service.

6.1.1 Call control

Call control is an association between one or more users and the network to control the set-up, release, modification and maintenance of a call (supported by sets of connections). Call control is used to maintain the association between parties and a call may be supported by 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 transferring control information in a single call/connection protocol;

– separation of information and state machines by providing separate protocol encapsulation 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, messages are exchanged between users and the network to negotiate the call characteristics. The exchange of messages between the calling party and the network is known as an outgoing call. The exchange of messages between the network and the called party is referred to as an incoming call.

Active

During this phase, user data can be transferred on the associated connections. The call parameters may be modified (e.g., the bandwidth of the supporting connections may be changed; new parties may be added to a point-to-multi-point call, where this type of call is supported).

Release

During this phase, 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 party or the network may release the call.

6.1.2 Call admission control

Call admission control is a policy 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 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.

6.1.3 Connection control

Connection control is responsible for the overall control of individual connections. This includes the set-up, release and modification procedures associated with a connection and the maintenance of the state of the connection.

6.1.4 Connection admission control

Connection admission control is 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 involves checking if there are free resources available. For packet switched networks (e.g., MPLS-TP), 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.

6.1.5 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. The call and connection coincide at domain boundaries.

7 Transport resources and their representation

The transport network is a large, complex network with various components, and an appropriate network model with well-defined, technology agnostic, functional entities is essential for its design, control, and management. The transport network can be described by defining the associations between points in the network. The resultant logical network topology allows the separation between logical connections and the physical routes and resources used.

7.1 Transport functional architecture

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. As described in [ITU-T G.805] and [ITU-T G.800], in order to simplify the description, the functional architecture utilizes the concepts of layering and partitioning within each layer network in a manner that allows a high degree of transport network recursion.

– a layer network describes the generation, transport and termination of a particular characteristic information (CI). Layering enables decomposition of a transport network into a number of independent transport layer networks. There is a client/server relationship between each of these layer networks where the client refers to the signal being carried, and the server refers to the layer network providing its transport. The client/server paradigm is recursive because any particular server layer could itself be a client of another server layer.
– partitioning is the division of a larger subnetwork into disjoint subnetworks that are interconnected by links. Because the model requires partitions to be nested, partitioning is also recursive.

The components of the transport network architectural model can be divided into three groups: topological components, transport processing functions, and transport entities.

1) topological components: Provide a description of a transport network in terms of the relationship between sets of forwarding points (FPs) within a layer network. These encompass, for example, layer network, subnetwork, link (including transitional link), access group.

2) transport processing functions: Are used to model the processes implemented in equipment that manipulate the information that is being transferred across the transport network. These include, for example, adaptation, (trail) termination, layer processor, forwarding.

3) transport entities: Provide the means to transfer information across the transport network between forwarding points. Transport entities are configured within topological components. These encompass, for example, forwarding relationship, subnetwork transport entity, connection, link connection.

The optical transport network (OTN) has both digital layer networks and a media layer. The control of optical signals and the media layer must take into account the limitations imposed by the accumulation of analogue impairments within the media. Many of these impairments and the magnitude of their effects are associated with particular technological implementations and the network topology. Annex A provides some clarification on the control of the media layer.

7.2 Domains

ASON control and SDN controller deployments will occur within the context of network 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 referred to is the generalization of the [ITU-T G.805] definition of administrative domain and the Internet administrative regions (e.g., autonomous systems) 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 scope (or boundary) of a domain is defined for a particular purpose, and domains defined for one purpose need not coincide with domains defined for another purpose. Domains that have been defined for the same purpose are restricted in that they do not overlap; however, they may:

– fully contain other domains that have been defined for the same purpose;

– border each other;

– be isolated from each other.

Examples of domains include administrative domains, control domains, routing domains, and recovery domains. Resources may be shared among domains of the same type; an example of this is the shared resources in a virtual network (VN), as described in clause 7.4.

The current assumption is that a control domain must be contained in an administrative domain. Investigation of a peer relationship between control domains that are in different administrative domains is for further study.

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

captures the common semantics of the information exchanged between MC component distribution interfaces, while allowing for different representations inside the domain.

Generally, a control domain is derived from a particular MC component type, or types, that interact for a particular purpose. For example, routing 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 MC components associated with a layer network.

Collections of MC components and functions may interact at reference points. These reference points include [ITU-T G.7703] UNIs, I-NNIs and E-NNIs, and [ITU-T G.7702] CPIs. Control domains border each other at these reference points.

7.3 Control view of transport resources for connection management

Connectivity is the basic service provided by a transport network. Connection management functions include: path computation, connection creation, connection modification, connection teardown, and configuration and activation of operations, administration and maintenance (OAM) and survivability mechanisms. These connection management functions could be operated under instructions from an SDN controller, management system, or ASON control.

The underlying transport resources used to support aspects of connection management to the SDN controller and ASON control functions are represented by a set of entities, termed subnetwork point (SNP) and subnetwork point pool (SNPP). The SNP and SNPP entities are organized into routing area (RA), subnetwork and link topological constructs which represent the view of the transport resources as seen by the SDN controller and ASON MC components from a connection management perspective. Figure 7-1 illustrates the relationship of the SNP to the transport resources described in [ITU-T G.800] and, the entities that represent these resources from the perspective of network management (as described in [ITU-T M.3100]).

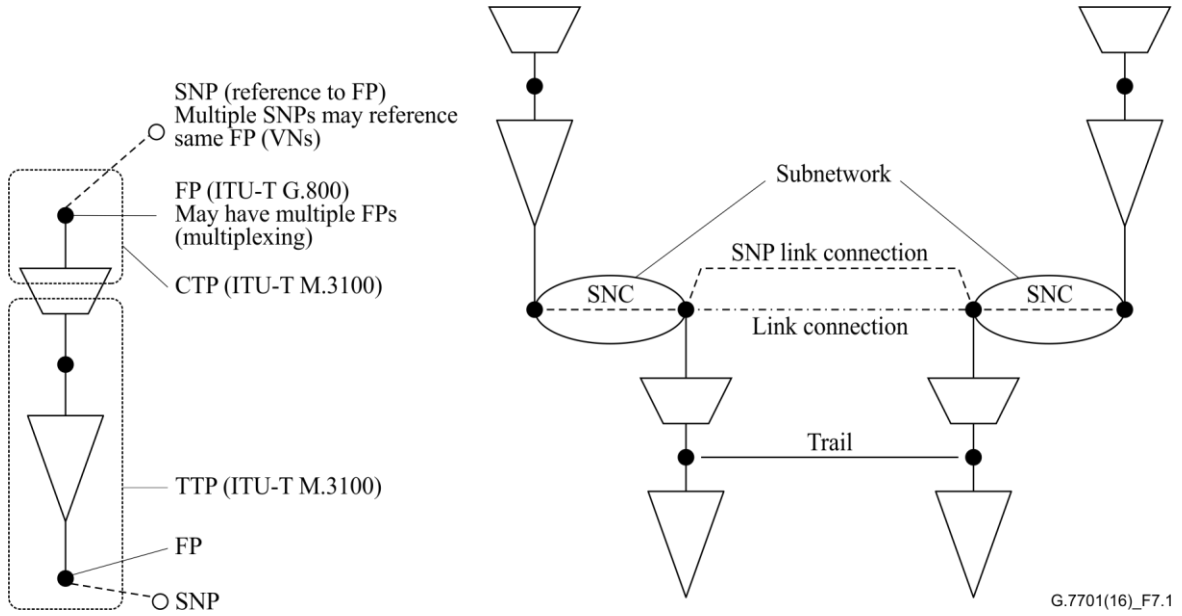


Figure 7-1 – Relationship between architectural entities in the control-management continuum

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) SNPs at the boundary of the same subnetwork. This is referred to as a subnetwork connection (SNC).

An SNP may also be grouped with other SNPs for the purpose of routing. This is an SNPP and has a strong relationship with link ends; 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 subdivided 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.

Figure 7-2 illustrates the relationship between the transport resources described in [ITU-T G.800] and the entities that represent these resources from the perspective of network management as described in [ITU-T M.3100] and [ITU-T G.7711], respectively.

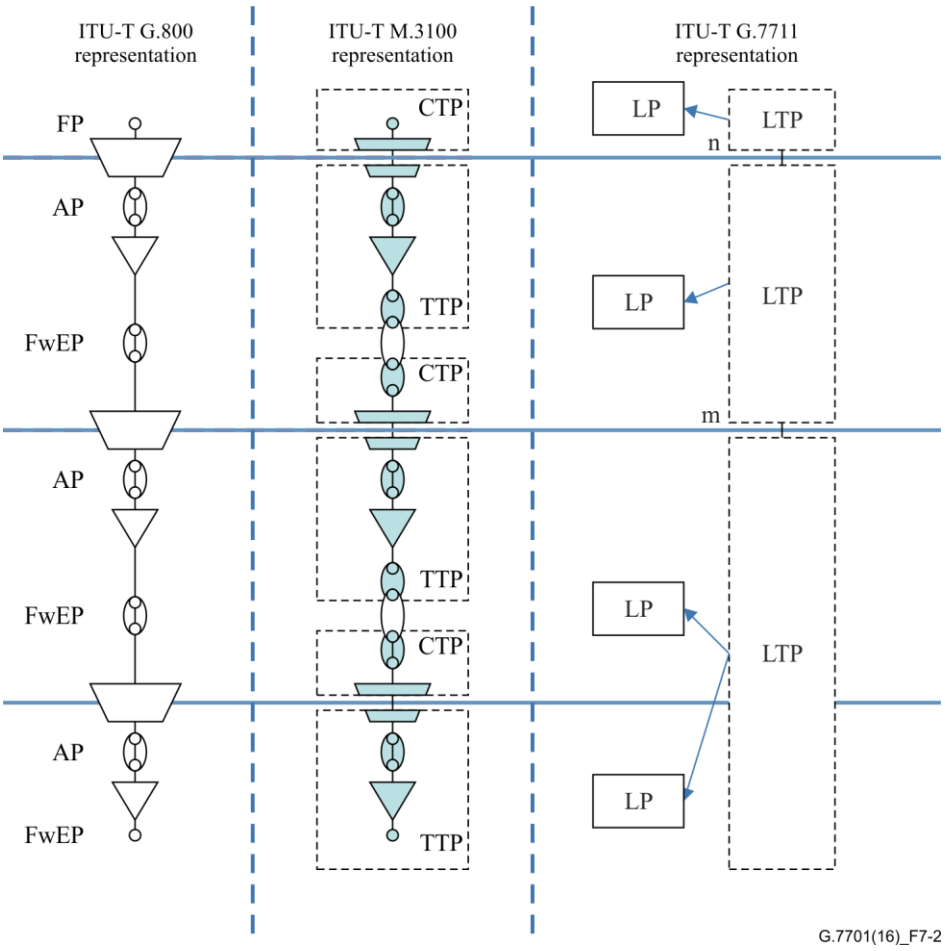


Figure 7-2 – Relationship of ITU-T G.800 architectural entities with their representation from a management perspective

As may be seen from Figure 7-1 and Figure 7-2 (which provides a simplified version of Figure 5-2 of [ITU-T G.7711]), the logical termination point (LTP) is used to represent transport resources and the FP, forwarding end point (FwEP), and SNP reference points.

7.4 Virtualization

An abstraction is a representation of a resource that represents selected characteristics, while hiding or summarizing characteristics irrelevant to the selection criteria. An abstraction may use an identifier from a name space that is different from the name space of the resource that is being represented. A virtualization is a set of abstractions that are a subset of the network whose selection criterion is dedication of resources to a particular client or application. A VN is an abstraction of a designated subset of [ITU-T G.800] layer network resources.

A connection routed in a VN can only use the SNPs associated with that VN.

An FP can be assigned and allocated to only one VN; in this case the FP is represented by a single SNP (in one VN).

An FP can be assigned (but not allocated) to multiple VNs; in this case the FP is represented by multiple SNPs (one in each VN). Connectivity on a link that is shared between VNs is modelled by creating a (potential) SNP for each of the shared FPs in each VN. When an FP is allocated to a particular SNP in one VN, the SNPs referencing the same FP in other VNs become busy. This is illustrated in Figure 7-3 which shows an example of two VNs, each with two SNPs. The resources support three FPs, FP_1 is assigned to VN2 and is represented by $SNP2_1$, FP_3 is assigned to VN1 and is represented by $SNP1_1$, FP_2 is shared by VN1 and VN2, FP_2 is represented by $SNP1_2$ in VN1 and $SNP2_2$ in VN2.

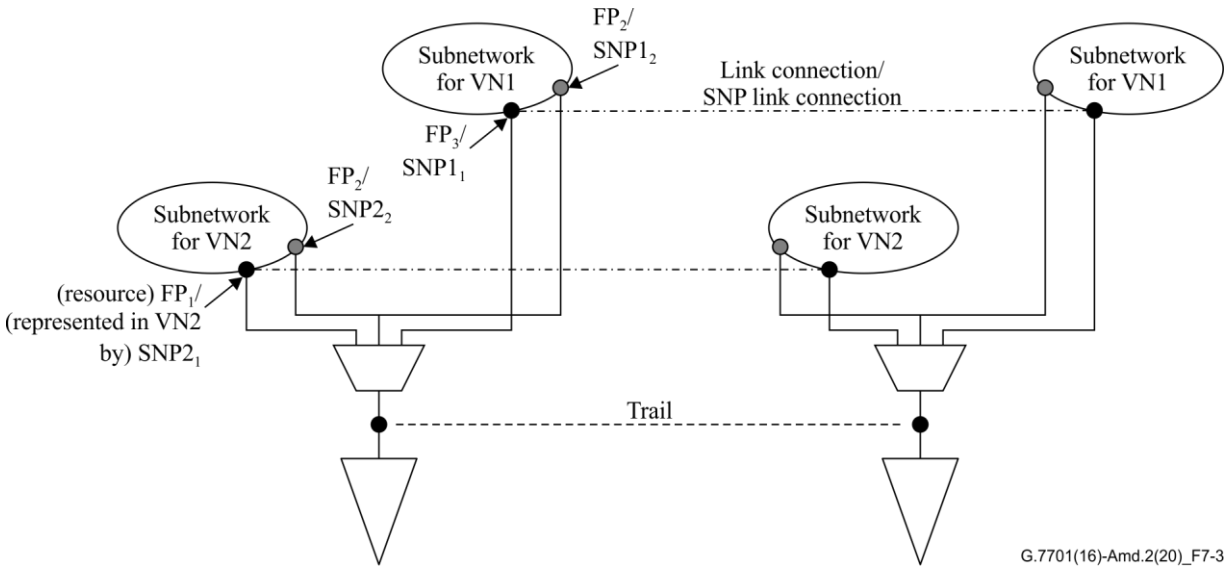


Figure 7-3 – Allocation of link resources between VNs

During the operation of the network if FP_2 is allocated to VN1, $SNP1_2$ becomes available while $SNP2_2$ (in VN2) becomes busy.

A resource label is the information required to distinguish a communication within the context of other communications [ITU-T G.800] (examples of a resource label are; a multiprotocol label switching (MPLS-TP) label in a CO-PS network or; a time slot in a CO-CS network). The SNP references an FP that has a resource label. In the case where the resource label does not carry a resource reservation (e.g., packet switching, see clause 8.3.4), the connection request must include an explicit resource reservation.

7.5 Multi-layer aspects

In a multi-layer network, the topology and connectivity of the underlying server layer is not fully visible to the client layer, aspects of the server layer are encapsulated and presented to the client layer network. The server layer topology may be represented as a client layer SNPP link, or as client layer subnetworks interconnected by SNPP links. 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 define the availability of underlying server layer resources to the client layer.

7.5.1 Representation as SNP link connections

The configuration of the server layer 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. If the resources supporting a SNP link are dedicated to the client, then the SNP link is active. If the resources supporting the SNP link are shared then the SNP link will be potential (see clause 7.4).

This 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 MC component. At this time, the client layer connection 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 configuration of the client layer connection 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.

7.5.2 Representation as a set of SNPP links and subnetworks

The server layer may be configured to allow the client layer MC components more flexibility in the selection of the server layer resources to satisfy a connection request. To accomplish this, a subset of the server layer resources are represented to the client layer as SNPP links and subnetworks. An example of the server layer flexibility and the corresponding client layer flexibility is shown in Figure 7-4. This enables the routing controller in the client to know that a set of client SNPs can be reached through a common server layer.

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 Figure 7-4, the dashed elements (links and subnetwork) show the representation of the server layer resources to the client layer. The server layer resources (adaptation, termination, links and subnetworks) are shown with solid lines. The client and server layer SNPs are represented as solid black and white circles.

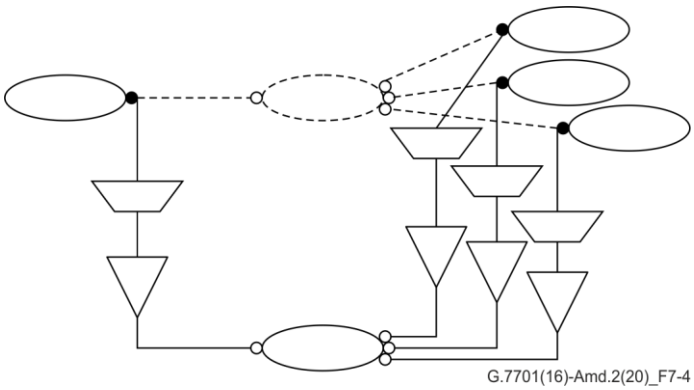


Figure 7-4 – Relationship between client and server architectural elements

As described in clause 7.5.1, the SNP link connections may be potential, if the server layer resources have not been allocated for exclusive use to this client. 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 and subnetwork connections are created, and processing continues as above. The server layer subnetwork connection will support one or more client layer subnetwork connections.

7.5.3 Multi-layer routing topology

In addition to the single layer topology representation in clause 7.5.2, 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. MC 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 7-5 illustrates the [ITU-T G.800] representation and the corresponding multi-layer routing topologies.

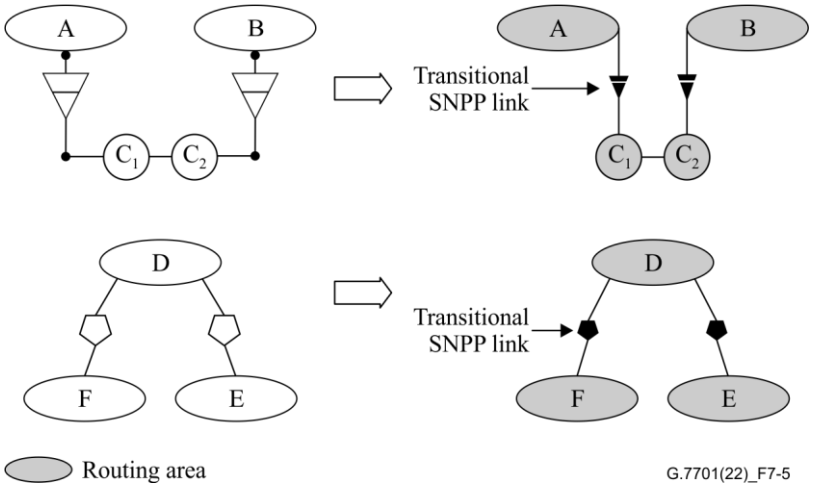


Figure 7-5 – Multi-layer routing topology representations

A multi-layer topology is illustrated in Figure 7-6 on the left.

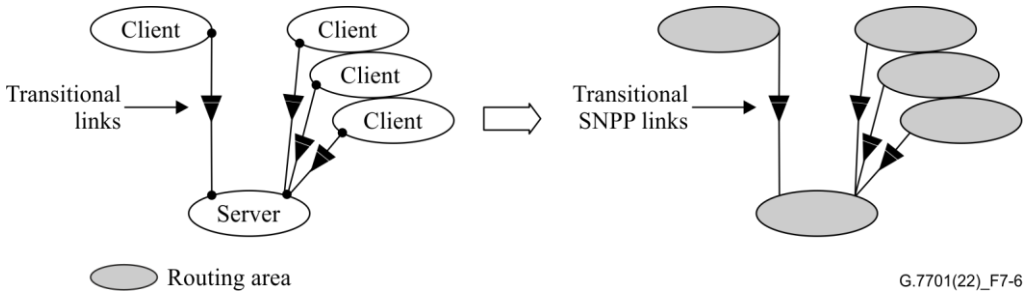


Figure 7-6 – Multi-layer routing topology representation

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 multi-layer 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 multi-layer topology is shown in Appendix I.

NOTE – Use of transitional links for a 1:1 adaptation is described in this clause.

7.6 Interlayer client support

In transport networks, resources from multiple layer networks may be present. For example, resources at the edge of the network may support the adaptation of service into a lower order ODU whilst resources in the core of the network may only provide flexibility at the high order ODU; or adaption of an Ethernet service into an ODU at the edge. 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 access group containers.

There are two solutions to this problem. Either client layer topology may be created from server layer connections as described in clause 7.5 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 8.3.1) in different layer networks are used in the second solution. This interlayer interface enables an association between calls in a client/server layer relationship. This association can recur to mirror a set of "stacked" adaptations. That is, the NCCs recur with [ITU-T G.800] layer networks. NCCs in different layer networks 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 in an ASON architecture. In an SDN architecture, it may be in several controllers with client layer scope and in one controller with server scope, or vice versa. 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 recurs upward and creates a link connection at each of the affected client layers.

The interlayer NCC relationship may occur at points other than where access group containers are attached to the client layer network. In Figure 7-4, a call traverses a client subnetwork first before being supported by a server layer subnetwork. In Figure 7-7, 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.

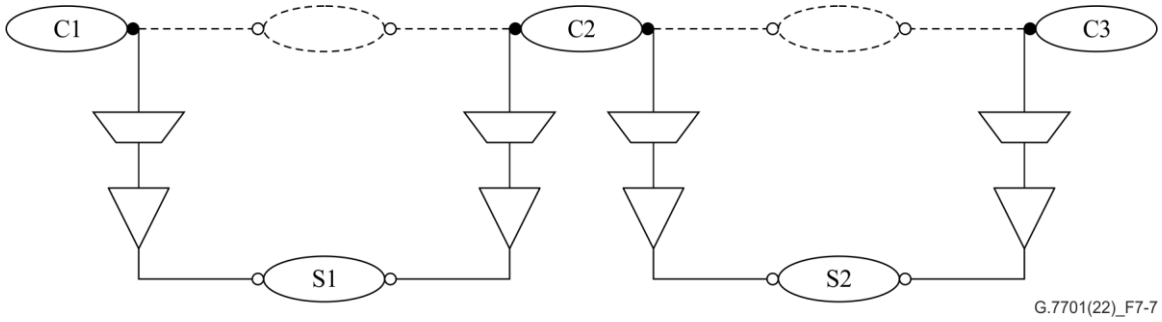


Figure 7-7 – Non-contiguous client layer subnetworks

7.7 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. In this arrangement, an NCC to NCC call segment is supported by a complete call with calling/called party call controllers. Figure 7-8 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 connection within Carrier B. The connection returned by the management and control of Carrier B is joined with the connections established in the Carrier A subnetworks.

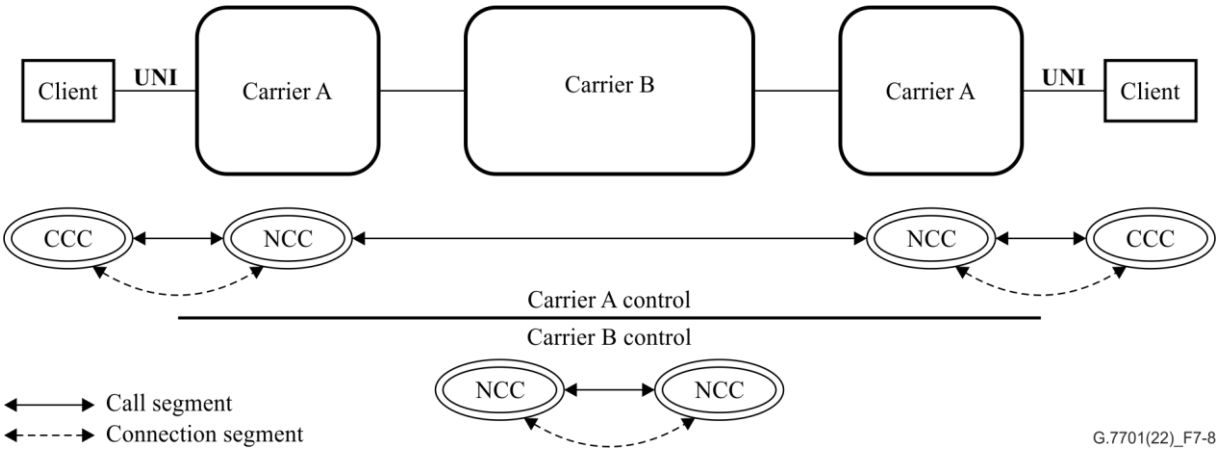


Figure 7-8 – Calls supported by calls in the same layer

NOTE 1 – In an ASON architecture this would not use an E-NNI between carrier MC components in each carrier as there would be no RC-RC interaction. Interlayer NCC calls specify the SNPs to be connected in the server layer. Thus, an RC-RC association is not present with the interlayer NCC pair to make the server SNPs visible in the client NCC, and the boundary between the two subnetworks is not an ENNI reference point.

NOTE 2 – Hierarchical SDN controllers could achieve the same result using interaction type 3 described in clause 15.3 of [ITU-T G.7702].

7.8 Mapped server interlayer relationships

When a client layer CI is mapped to a server layer as described in clauses 7.5 and 7.6, several types of arrangements can exist. They include client/servers in 1:1, 1:n and n:1 relationships. The ratio refers to the number of connection points in each layer.

7.8.1 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.

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.

7.8.2 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 7-9 illustrates the latter case. The Ethernet call is mapped to a VC-4-2v virtual concatenation (VCAT) call. The VCAT call is related to multiple server layer VC-4 calls. The Ethernet/GFP layer to VCAT relationship is 1:1. The relationship of Ethernet/GFP layer NCC to VCAT NCC is 1:1, and the relationship of VCAT layer NCC to VC4 server layer NCC is 1:n.

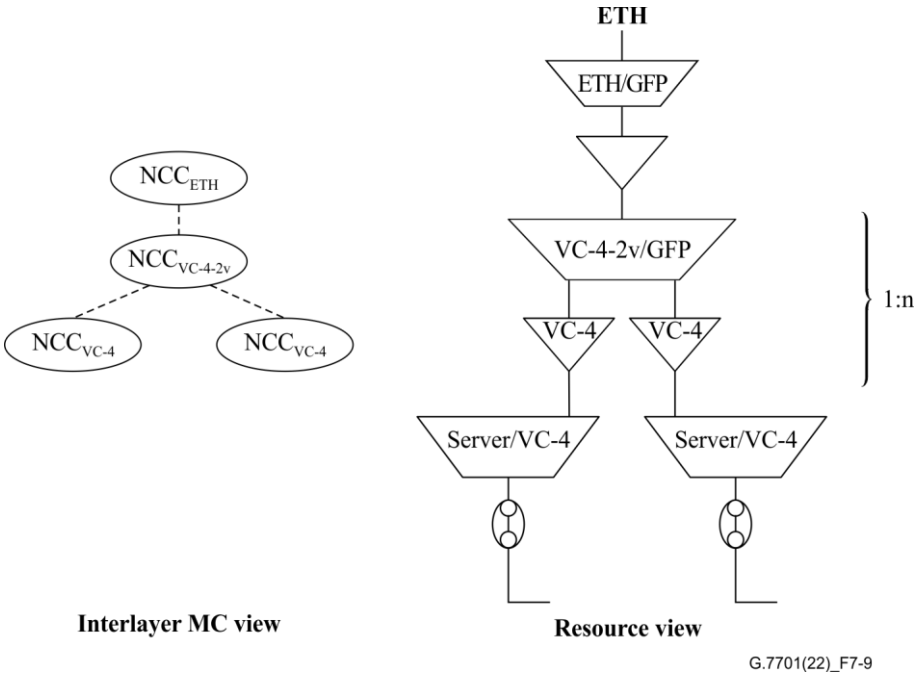


Figure 7-9 – Example 1:n mapped server

It is necessary for the client layer NCC (the VC-4-2v layer) to be aware of 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.

7.8.3 n:1 relationship

In an n:1 relationship, the multiple client communications are supported by a connection in the server layer. This is supported either by one client NCC supporting multiple connections or multiple NCCs in the client layer associated with one server layer NCC.

The example below in Figure 7-10 illustrates the latter case. Each Ethernet call is mapped to an MPLS-TP PW call. Multiple MPLS-TP PW calls are mapped to a server layer MPLS-TP LSP. The MT/ETH layer to MPLS-TP PW relationship is 1:1. The relationship of MT/ETH layer NCC to MPLS-TP PW layer NCC is 1:1, and the relationship of MPLS-TP PW layer NCC to MPLS-TP LSP layer NCC is n:1.

It is necessary for the server layer NCC (MPLS-TP LSP layer) to be aware of the call parameters of the client layer as it must ensure that it has sufficient resource to support multiple client layer calls as well as what their aggregate characteristics are. The client layer is not required to possess knowledge of the server layer call parameters but should inform the server call if there are changes in the client layer call.

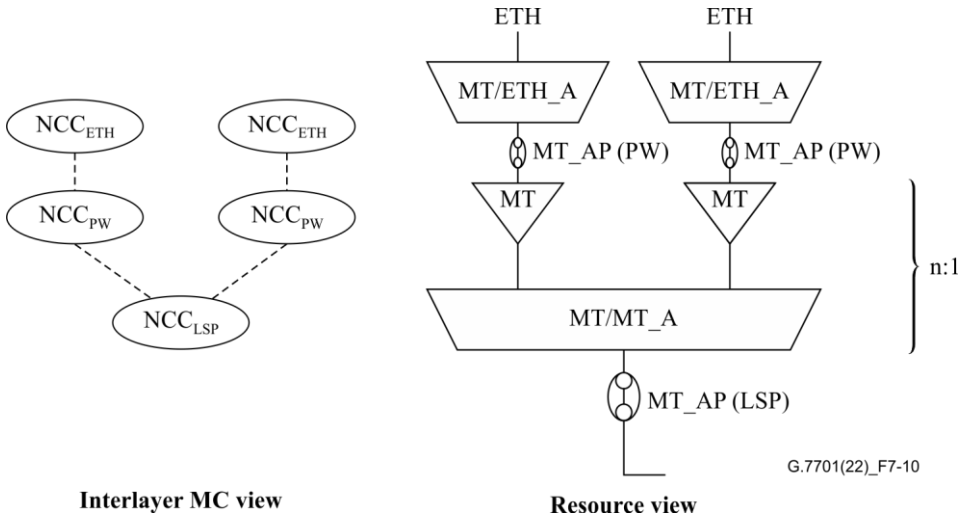


Figure 7-10 – Example n:1 mapped server

The following issues should be considered when adopting n:1 relationship:

- The first client layer network connection consumes n × its capacity in the server layer, this capacity is only use when additional client communications are supported.
- Adding capacity in a different location may make more efficient use of the server layer resources, based on the projected future traffic demands.
- A policy for removing server layer connections, that were established based on a client request, when the last client layer network connection is removed must be established.

These issues can be avoided if a planning process is used to predict when the client requires additional server capacity.

8 MC component approach

An MC component approach is used to express the architectures for ASON and for application of SDN to transport networks in a way that facilitates the construction of reasonable scenarios.

MC components are identified in such a way that the most commonly used distributions of functionality are supported. Interfaces are exposed according to when there is a desire to distribute MC components among platforms, which is implementation specific. Where this is the case, the primitives exchanged between MC components are combined and carried over an external protocol.

It should be noted that the architecture is not solely comprised of MC components and their interactions, but also assumes the presence of other key conceptual constructs. A resource database (RDB) is a logical entity that holds information used by MC components and it is assumed to be present as needed to support MC components. The RDB is used by various distributions of MC components and may be used to exchange information between MC components.

8.1 Notation

In unified modelling language (UML) [b-UML], a component is described in section 11.6as a replaceable modular unit with defined interfaces that specify the services offered to other components and the services that it requires from other components.

This Recommendation reuses the concept of the UML component. An MC component is defined as "an element that is a replaceable part of a system that conforms to and provides the realisation of a set of interfaces". An MC component represents an abstract entity rather than a piece of implementation code. Thus, MC components represent logical functions rather than implementations. The component approach allows the MC architecture to be described in terms of MC components. This Recommendation describes the common aspects of MC components, any required SDN or ASON specific extensions are described in [ITU-T G.7702] and [ITU-T G.7703] respectively. To take advantage of the commonality of SDN and ASON these extensions are described in a way that is analogous to inheritance and composition used in UML.

MC component interface: 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 an MC component they are represented by an output interface with output parameters and no return parameters.

Interface definitions are presented in the form of a table, an example of which is provided in Table 8-1.

Table 8-1 – Generic component 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.

MC Component: In this Recommendation, MC components are used to represent functions, rather than instances of implementation code. They are used to construct scenarios to explain the operation of the architecture. An MC component is represented as a rectangle with tabs. This is illustrated in Figure 8-1.

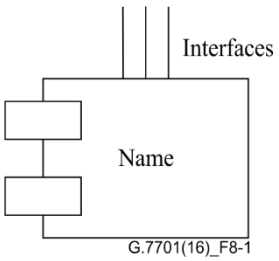


Figure 8-1 – Representation of a component

In addition to the input, output and notification interfaces described above, each MC component may provide an additional set of interfaces to allow configuration of policies that affect the internal behaviour and obtaining information about the resources supporting the MC component. These interfaces are not mandatory, they are provided on specific instances of MC components when necessary. The results from monitoring of MC component operation are reported via the notification out interface. These interfaces are used to create, modify, delete or monitor MC components as described in [ITU-T G.7716] and [ITU-T G.7718]. These additional interfaces are not described in this Recommendation.

MC components are not assumed to be statically distributed.

Details of the input, output and notification MC component interfaces are defined based on the operations that the MC component is intended to perform and are, therefore, MC component specific. When interfaces on MC components are described, only the different interface types are specified. MC components may support one or more instances of an interface. Any required SDN or ASON specific extensions of an interface are described in [ITU-T G.7702] and [ITU-T G.7703] respectively.

MC components are collected into a SDN MC system or an ASON MC system. Parameters are transferred directly between MC components in the same MC system. MC systems are interconnected via the CCN. The protocol controller described in clause 8.3.9 provides an external interface that attaches to a reference point on the CCN and provides authenticated, secure, and reliable transfer of parameters across the CCN. A SDN or ASON MC system consists of a collection of one or more protocol controllers and one or more other MC components or two or more MC systems.

8.2 Policy

Policy is defined as the set of one or more rules that define the action that an MC component takes in response to a set of conditions (e.g., parameters or messages) presented at an interface. More than one policy may be applied (sequentially) to these input conditions to determine the final action taken by the MC component. Similarly, the action taken by an MC system in response to a set of input conditions (parameters) contained in a message, is the determined by the policies applied (sequentially) by each of the MC components that process the input conditions.

The policies of an MC component define the internal behaviour of that MC component i.e., the processing applied to the conditions present on an input interface to generate parameters or messages on an output interface. If the input conditions do not satisfy the policy the MC component may not provide any output parameters, or it may output an exception.

Each policy may be configured using a configuration interface (described in clause 8.1) or a policy may be provided as a parameter presented at an input interface. The ability to modify a particular policy may be controlled by a policy applied to the configuration interface. This provides control over the aspects of policy (system behaviour) that are fixed and the aspects that may be modified.

As described in clause 8.1 the external interface(s) to an MC system are connected to reference points on the CCN. Each external interface carries one or more abstract interfaces. The policies are invoked on an external interface by a policy enforcement point (PEP) (defined in [ITU-T X.509] as the point where policies are enforced). These policies may include for example, encryption, authentication and authorization of the messages. The PEP rejects (discards) any messages that do not conform to the policies.

The external interfaces provide reference points (with multiple abstract interfaces), policies are applied to the individual abstract interfaces that cross the reference point.

The protocol controller component (described in clause 8.3.9) implements the PEP and multiplexes one or more abstract interfaces into a single external interface.

Policy, as other aspects of the system, may be distributed. The PEP, the point where the policy decisions are enforced and the policy decision point (PDP), (defined in [ITU-T X.509] as the point where policy decisions are made) may be implemented by the protocol controller, or the PDP may be in a different MC system. This distribution decision depends on the actual policy. For example, for performance reasons the PDP may be within the protocol controller (e.g., for encryption), while for security reasons the PDP may be elsewhere (e.g., for authentication).

8.3 Common MC components

The component architecture for the application of control functions to transport networks is described in this clause. A summary of the primary function of each MC component is first provided, followed by descriptions of MC component interfaces and detailed operations, respectively.

The connection controller (CC), hierarchical call controller, and RC are all MC components. The VN context includes the name space that MC components use to operate on VN resources.

Protocol controllers are provided to take the primitive interface supplied by one or more MC components, and multiplex those interfaces into a single instance of a protocol. In this way, a PC absorbs variations among various protocol choices, and the MC component architecture remains invariant. One or more PCs are responsible for managing the information flows across a reference point.

8.3.1 Call controller components

Calls are controlled by means of call controllers, which are able to recur either in a hierarchy, horizontally (as peers) or both.

There are two types of call controller components:

1) a calling/called party call controller (CCC): This is associated with an end of a call and may be co-located with end systems or located remotely and act as a proxy on behalf of end systems. This controller has two roles, one to support the calling party, the other to support the called party.
2) a network call controller (NCC): This 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.

The calling party role interacts with the called party role via one or more intermediate NCCs.

Network call controllers are instantiated at policy boundaries (e.g., at reference points or aggregated interfaces).

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

NCC-NCC relationships may occur in peering and hierarchical relationships. Four cases of these are:

1) Peering. NCC-NCC interactions are in the same layer network and form a chain which is a form of the [ITU-T G.7703] cooperative federation model.

2) Hierarchical in the same layer but with different (routing) levels of resource abstractions in scope. NCCs are arranged in a hierarchy of MC systems whose resources are in the same layer network. MC systems at the bottom of the hierarchy have the FP name space of the resources. MC Systems above those use SNP/SNPP name spaces for abstractions of the resources. Client NCCs make requests of server NCCs in order to complete calls in their scope.

3) a) Hierarchical with different resources between MC systems. Resources are in different layer networks. NCCs are arranged in a hierarchy and a pair of NCCs in MC systems, each with resources in different layer networks, have an association for the purpose of call setup between layers. Typically, this corresponds to a transitional link in a multi-layer topology.

b) Hierarchical with different resources between MC systems. Resources are in the same layer network. In this case client calls are supported by calls in a different routing level but at the same network layer. MC systems with the client calls have a different set of resources as MC systems containing the server calls.

8.3.1.1 Calling/called party call controller

The role of CCC 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 8-2. The calling/called party call controller component is illustrated in Figure 8-2.

Table 8-2 – 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

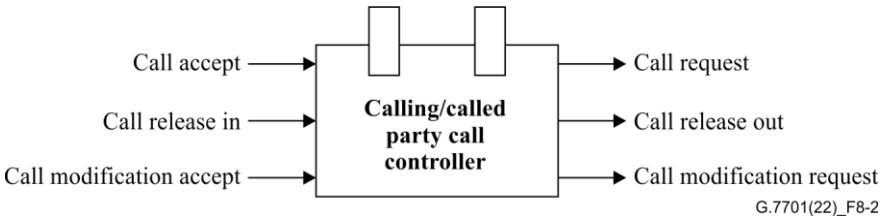


Figure 8-2 – Calling/called party call controller component

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.

8.3.1.2 Network call controller

The role of NCC component is:

– processing of incoming call requests;
– generation of outgoing call requests;
– generation of call termination requests;
– processing of call termination requests;
– processing of call modification requests;
– translation from call source and destination identifiers (which are boundary resource identifiers (BRI)) to identifiers from the control name space (i.e., SNPs, SNPPs) via directory request;
– call admission control based on validation of call parameters, user rights and access to network resource policy;
– state management of client calls and itself;
– application of policies such as quality of service (QoS) and recovery (e.g., use of protection or restoration);

This component has the interfaces provided in Table 8-3 and illustrated in Figure 8-3.

Table 8-3 – Network call controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Call request accept	Boundary resource identifier	Confirmation or rejection of call request
Network call coordination in	Boundary resource identifier	Confirmation or rejection
Call release in	Boundary resource identifier	Confirmation of call release
Client NCC coordination in	Optional client call parameters, optional client context or client layer identification, boundary resource identifiers	A pair of SNPs in the view of the client NCC
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

Output interface	Basic output parameters	Basic return parameters
Call indication	Boundary resource identifier	Confirmation of rejection of call request
Connection request out	Boundary resource identifier	A pair of SNPs
Network call coordination out	Boundary resource identifier	Confirmation or rejection of call request
Directory request	Boundary resource identifier	SNP, SNPP
Call release out	Boundary resource identifier	Confirmation of call release
Client NCC coordination out	A pair of SNPs in the view of the client NCC	Confirmation or rejection of use
Server NCC coordination out	Optional call parameters, context identification, boundary resource identifiers	A pair of SNPs
Call modification indication	Call name, parameters to change	Confirmation or rejection of call modification
Notification out	notification	

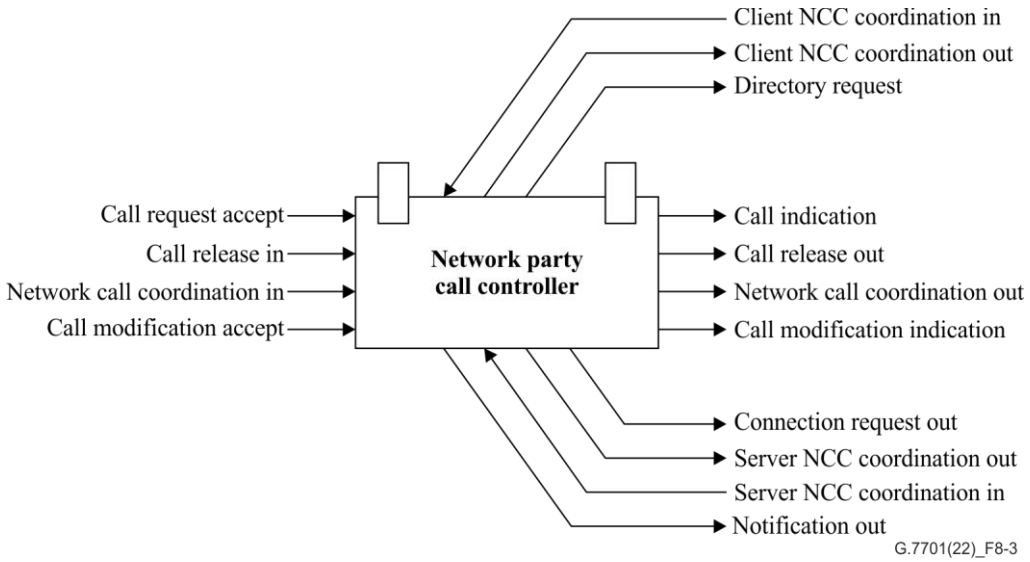


Figure 8-3 – Network call controller component

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 setup request. It is used in the cooperative model for peer call controllers.

Call indication: Call requests are forwarded to NCCs as calls are progressed. This interface also confirms or rejects the outgoing call setup request. It is used in the cooperative model for peer call controllers.

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

Directory request: This interface is used to translate a boundary resource identifier (BRI) into an SNP/SNPP identifier.

Network call coordination: This interface is used for network-level call coordination in the cooperative model between NCCs.

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

Client NCC coordination in: This interface is used in NCC hierarchies to accept a request from the calling NCC for a pair of SNPs. The called NCC is provided with source and destination identifiers in its view in order for it to provide a network connection for use by the calling NCC. The NCCs can be in the different layers or in the same layer with a hierarchical arrangement. For NCCs in different layers, SNPs in the view of the calling NCC that are supported by an adaptation to the network connection in the server layer are returned, such as the case described in clause 7.6. For NCCs in the same layer, SNPs are returned that are used by the CC associated with the calling NCC, which is in a hierarchical arrangement, such as the case described in clause 7.7. This interface is also used to release or modify the use of the SNP pair. The called NCC returns the result of the action.

Client NCC coordination out: This interface is used in NCC hierarchies to present a pair of SNPs to the calling NCC. The NCCs can be in a different layer or in the same layer with the hierarchical arrangement. The SNPs are supported by an adaptation to a network connection when the NCCs are in the different layers. The calling NCC indicates whether or not it accepts this resource. This interface is also used to present a modified or released SNP pair. The calling NCC returns the result of the action.

Server NCC coordination out: This interface is used in NCC hierarchies to request a pair of SNPs that can be used by the call to transfer CI. The NCCs can in the different layers or NCCs in the same layer with a hierarchical arrangement as described in clause 7.8. For a called NCC in a different layer, the input SNPs are identical to the return parameters of the connection request out interface except that a network connection in the layer of the called NCC is not assumed to be created. For a called NCC in the same layer, the requested SNPs are identical to the return parameters of the connection request out interface. This interface is also used to release or request modification of the use of the SNP pair provided by the called NCC. The called NCC returns the result of the action.

Server NCC coordination in: This interface is used in NCC hierarchies to present a pair of SNPs from the called NCC. The NCCs can be in a different layer or the same layer with a hierarchical arrangement. The SNPs may be accepted or rejected by the calling NCC. This interface is also used to present a modified or released SNP pair. The calling 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. It is used in the cooperative model for peer call controllers.

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. It is used in the cooperative model for peer call controllers.

Notification out: This interface is used to inform the notification component of a call related notification.

The role of call admission control in the calling party NCC 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 NCC, 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 NCC 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.

Using the Call request accept and Call indication interface**,** a call is progressed between CCC and NCCs forming call segments. The Network call coordination in and Network coordination out interfaces are used as a call is progressed between NCCs forming call segments. A chain of NCCs can be formed that result in a chain of call segments. For the hierarchical NCC case the Client/Server NCC Coordination in/out interfaces are used. These returned pairs of SNPs represent a connection and allow the calling NCC to gather both call and connection segments.

8.3.2 Connection controller component

The CC is responsible for coordination among the link resource manager (LRM), RC, and both peer and subordinate connection controllers for the purpose of the management and supervision of connection setup, release and the modification of connection parameters for existing connections. This component has the scope of a single subnetwork, and provides the abstract interfaces to other MC components given in Table 8-4. The connection controller component is illustrated in Figure 8-4.

Table 8-4 – Connection controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Connection request in	A pair of local SNP identifiers and optionally a route	A subnetwork connection
Peer coordination in	1) A pair of SNP identifiers; or 2) SNP and SNPP; or 3) SNPP pair; or 4) route	Confirmation signal
Connection correlation query	Original resource identifiers	Relevant state

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 identifiers	A subnetwork connection
Peer coordination out	1) A pair of SNP identifiers; or 2) SNP and SNPP; or 3) SNPP pair; or 4) route	Confirmation signal
Notification out	notification	

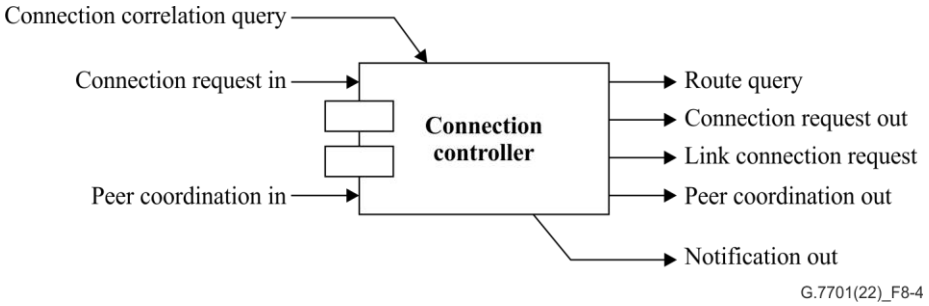


Figure 8-4 – Connection controller component

Connection request in: This interface is used to receive a connection request.

Connection request out: This interface is used to initiate a connection request to another connection controller component.

Peer coordination in: This interface is used to receive a connection request that is in progress. Specific or general topology identifiers are included to indicate points that the connection should traverse.

Peer coordination out: This interface is used to continue a connection request that is in progress. It is sent to a peer connection controller.

Connection correlation query: This interface is used to return to the notification component, the relationship between the resource that generated the notification and the connections which are supported by that resource.

Link connection request: This interface is used to request a link connection from LRM. It is also used at the bottom of hierarchical recursion and in the peer coordination process.

Route query: This interface is used to request a route from the routing controller. Some constituents of the route may be given in the request and the RC completes unresolved parts of the route.

Notification out: This interface is used to inform the notification component of a connection related notification.

Connection setup is performed in response to a connection request from, a network call controller, an enclosing scope connection controller, or a peer connection controller. In the case of hierarchical routing, there could be CCs in adjacent routing levels. The higher level CC divides the responsibility for the connection across the multiple lower routing levels. A higher level CC selects the source and destination SNPs and requests the connections from the CCs for the specific lower routing levels via the connection request in/out interface. The higher level CC concatenates the connections from lower level CCs together in an end-to-end manner. In all other cases, the peer coordination in/out interfaces are used. Component operation is the same in both cases.

When the CC has two link connections that are in sequence for a route, it creates the subnetwork connection (SNC) between the end of one link connection and the end of the other link connection. If the TAP component for those links is used, then the FP name space will be configured and the SNC will support information transfer. Otherwise, the SNP name space will be configured.

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 LRMs; however, the operation of the connection controller is invariant. Connection release is an analogous operation to connection setup, except the operations are reversed.

8.3.3 Routing controller component

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.

The role of the RC 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;

– topology (SNPs and their abstractions) information for control and management purposes.

Information contained in the RC (SNPPs, SNP link connections) enables it to provide routes across its routing domain. The RC may also receive topology updates from its peers, consisting of identifiers for adjacent nodes/subnetworks and their links, that may also include identifiers of specific peer subnetwork resources. RC information may be contained in the resource database. The resource database 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 may be provided.

The functional requirements of the RC include:

– topology management for intra-domain and multi-layer transport network;

– topology management for inter-domain and multi-layer transport network;

– path computation of intra-domain, inter-domain or multi-layer for transport network;

– abstraction of topology and resources of transport network.

The component architecture of RC is shown in Figure 8-5.

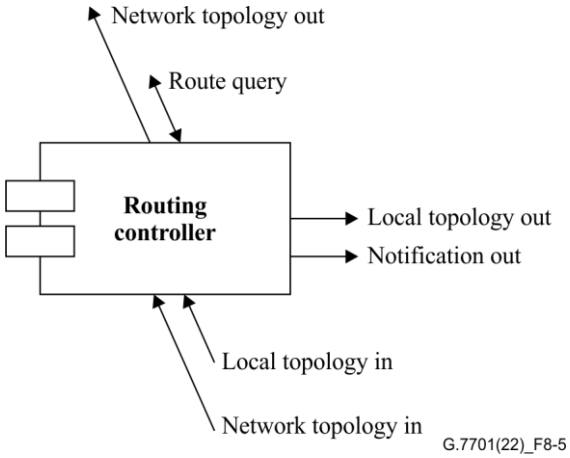


Figure 8-5 – Component architecture of routing controller

The input and output interfaces for RC are provided in Table 8-5.

Table 8-5 – Routing controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Route query	Unresolved route element, Route policy	Route
Local topology in	Local topology update	–
Network topology in	Topology update	–

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

Local topology interface in: The local topology interface in is used to maintain accurate information on the local topology and its state. This is the topology information that is within the domain of responsibility of the RC.

Local topology interface out: The local topology interface out is used to convey information on the local topology and its state.

Network topology interface in: The network topology interface in is used to retain accurate information on the network topology and its state. This is the topology information that is outside the domain of responsibility of the RC. Information from other RCs Local topology interface out and Network topology interface out are received on this interface.

Network topology interface out: The Network topology interface out is used to convey the information received from the Network topology interface in, or a summarized version thereof (further abstracted), and may also provide a summarized version of the local topology.

Route query interface: This interface accepts an unresolved route element and returns a route. Route queries may be either from the connection controller component (CC) or may be from another RC.

Notification out: This interface is used to inform the notification component of a routing related notification.

Depending upon the context, the RC component can receive local topology information from the LRM component or network topology information from an RC at a lower hierarchical level.

In the routing calculation process, RC can cooperate with other RCs to perform multi-domain and multi-layer route calculation.

8.3.4 Link resource manager component

The LRM component is responsible for the management of an SNPP link; this includes the assignment and unassignment of SNP link connections (to a connection), managing resource reservation, the configuration of policing and shaping functions via the TAP (if required), providing topology and status information. The SNPs and the associated resource (e.g., capacity) that are in the SNPP at each end of the link are provided by the TAP (see clause 8.3.6), the LRM also receives resource status information from the TAP. A 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). The LRM functions are shown in Figure 8-6.

Layer network using circuit switching with a fixed bitrate

The TAP supplies the LRM with the set of SNPs in the SNPP. When the TAP allocates an SNP, and associates it with a resource label, the FP and link connection are created; the resource label provides an implicit reservation of the link resource. The LRM may assign any of its SNPs to support a connection without further interaction with the TAP. The LRM can track the utilization of the SNPP link by tracking the SNPs that it has assigned. In general, the same SNP identifier and resource label are used for both directions of a bidirectional connection.

Layer network using circuit switching with a flexible bitrate (e.g., OTN with ODUflex)

The TAP supplies the LRM with the set of SNPs in the SNPP, the potential and allocated capacity, together with the capacity assignment policy. The FP and link connections are not created (by the TAP) until the LRM assigns an SNP and capacity to a connection. A LRM may request the TAP to modify (i.e., increase or decrease) the capacity used by the connection. The LRM advises the TAP when an SNP is assigned or unassigned, this allows the TAP to create or delete the FP. The LRM is responsible for the assignment of the allocated capacity within the constraints of the capacity reservation policy provided by the TAP. The LRM can track the utilization of the SNPP link by tracking the SNPs and capacity that it has assigned. In general, the same SNP identifier and resource label are used for both directions of a bidirectional connection.

Layer network using packet switching

The SNPP link information held by the LRM must include the admission control policy (e.g., amount of overbooking allowed for the committed information rate (CIR) and excess information rate (EIR)). When the LRM 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.

The TAP supplies the LRM with the set of SNPs in the SNPP, the potential and allocated capacity (CIR and EIR), together with the capacity assignment policy. The FP and link connections are not created (by the TAP) until the LRM assigns an SNP and capacity (CIR and EIR) to a connection. A LRM may request the TAP to modify (i.e., increase or decrease) the capacity used by the connection. The LRM advises the TAP when an SNP is assigned or unassigned, this allows the TAP to create or delete the FP. The LRM (via the TAP) must also configure the appropriate traffic policing and shaping functions. The LRM is responsible for the assignment of the allocated capacity within the constraints of the capacity reservation policy provided by the TAP. The LRM can track the utilization of the

SNPP link by tracking the SNPs and capacity (CIR and EIR) that it has assigned. Different SNP identifiers and resource labels may be used for each direction of a bidirectional connection.

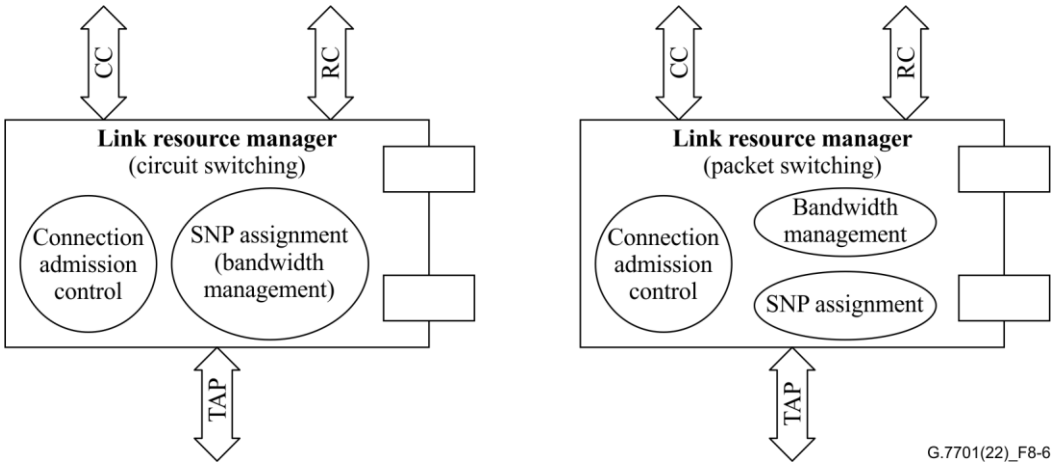


Figure 8-6 – Basic LRM functions for circuit switching and packet switching

To support controller-based restoration, the LRM must be aware of the fault reporting management (FRM) and fault event filtering (FEF) information [ITU-T G.7710] for each SNP. The call/connection set-up request can indicate to use the FRM and FEF information for restoration. The LRM assigns the SNPs to the call/connection and maintains the indication of using the FRM and FEF information for those assigned SNPs. TAP maintains the operational state of the SNP based on the filtered faults from the FRM function for the FP that supports that SNP. The LRM further updates and sends the changed link connection state to the CC.

The functional requirements of LRM include:

– the output of local topology (nodes and links), resources and abilities information of transport network;

– the status update of topology (nodes and links), resources and abilities information of transport network;

– link configuration, OAM configuration and maintenance;

– assist RC to update the status of topology of transport network;

– assist CC to update the status of connections of transport network.

The component architecture of LRM is shown in Figure 8-7 and the LRM component has the interfaces provided in Table 8-6:

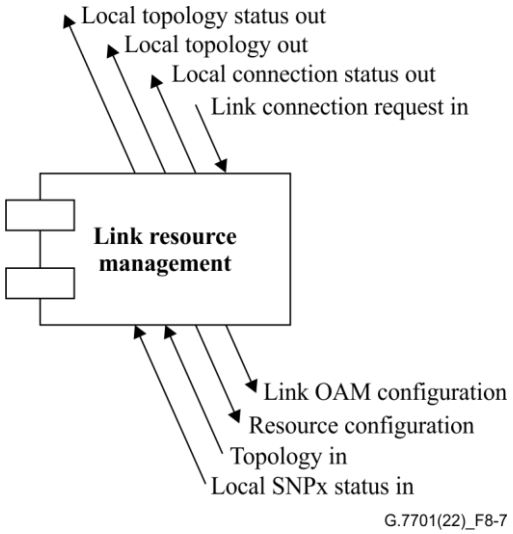


Figure 8-7 – Component architecture of LRM

Table 8-6 – Link resource management component interfaces

Input interface	Basic input parameters	Basic return parameters
Link connection request in	A link connection (an SNP pair)	Confirmation or rejection
Topology in	Local topology update	–
Local SNPx status in	Link SNPx status	

Output interface	Basic output parameters	Basic return parameters
Local topology out	Local topology update	–
Local topology status out	Network topology update	–
Link OAM configuration	OAM parameters	Confirmation or rejection of OAM parameters
Resource configuration	Resource information, FEF information	Confirmation or rejection of link information, FEF information
Local connection status out	Local connection status – operational state – administrative state – performance	

Link connection request in: This interface is used to receive the request from CC for a link connection between the associated SNP pair. This interface can be used to receive the indication for using the FEF information from the CC.

Topology in: This interface is used to receive the local topology updates from TAP.

Local SNPx status in: This interface is used to receive the SNPx status from TAP.

Local topology out: This interface is used to convey information on the local topology to RC.

Local topology status out: This interface is used to convey the information on the local topology status to RC.

Link OAM configuration: This interface is used to request TAP for the configuration of Link OAM.

Resource configuration: This interface is used to request TAP for the configuration of Link end information. This interface is used to request TAP to configure the FEF information for the associated SNPs.

Local connection status out: This interface is used to convey the information on the local connection status to CC, it also reports the state of SNPx.

8.3.5 Discovery agent component

The discovery agent (DA) component deals with transport resources that may not be assigned to an MC component. The federation of discovery agents operates in the transport resource name space. The federation has knowledge of FPs in the network, while a local DA has knowledge of only those FPs assigned to it.

The transport resource name space provides the native identifiers used by transport resources; this Recommendation assumes that a transport resource name space exists for the [ITU-T G.800] FPs in the layer network. SNP name spaces are mapped to those FPs and the mappings are maintained by TAP components. The TAP also maintains the relationship between the client forwarding point (cFP) and server forwarding point (sFP).

DAs discover the trail FP to FP relationship from which the link relationships from client FP to client FP are inferred [ITU-T G.7714.1]. Generally, multiple client FPs (cFPs) are associated with a trail FP in the server layer (sFP). Client FPs result from adaptation and if the adaptation is flexible, FPs in different layers may result. LRMs hold the corresponding SNP to SNP relationship for links in the SNP name space.

FP identifiers are only used in the context of DA and TAP components, all other MC components use SNP identifiers. The DA and TAP have access to the FP name space for the resources in their scope.

The DA uses the coordination in/out interface to communicate with adjacent DAs to obtain the sFP(s)-sFP(s) relationship between the local sFP and the remote sFP. The local cFP(s) and sFP and the remote cFP(s)-sFP binding relationship is provided by the local and remote TAP. With this information, the cFP-cFP relationship can be determined.

Figure 8-8 depicts actions of the DA used to establish the cFP-cFP relation.

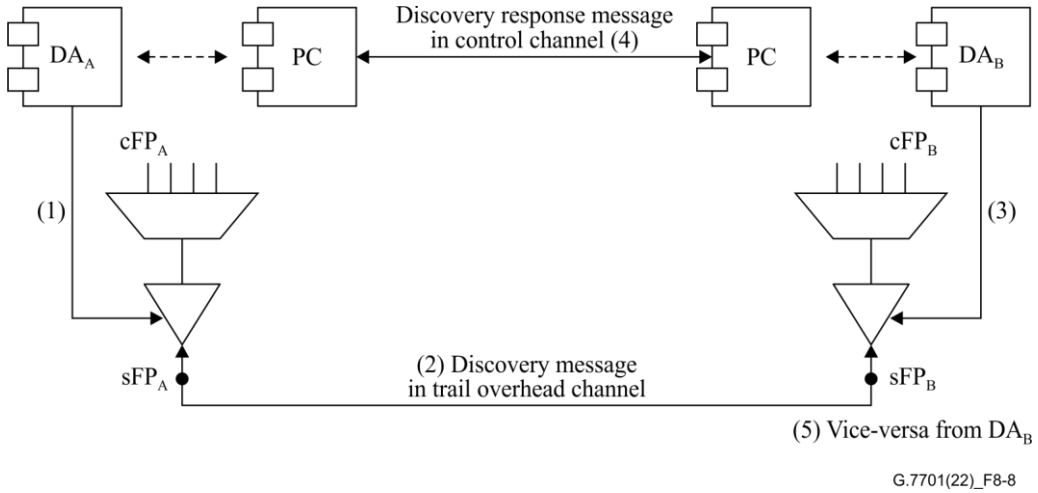


Figure 8-8 – Discovery agent actions

NOTE – The PCs are not needed in all cases.

The steps to establishing FP to FP relationships are:

1) DA_A triggers sFP_A to inject a discovery message (DM) that contains the sFP_A identifier and the DA_A identifier.

2) The discovery message is carried by the trail from sFP_A to sFP_B.

3) DA_B receives the discovery message from sFP_B, which contains sFP_A's identifier and the DA_A identifier.

4) DA_B sends a discovery response message to DA_A that provides the pair of server FPs (sFP_A and sFP_B) and the DA_B identifier.

5) The same discovery process (steps 1 through 4) is performed by DA_B in the reverse direction.

6) When the discovery processes in the two directions have completed, each DA locally compares the discovered sFP relationships in the forward and reverse directions to determine whether the relationships in the two directions A→B and B→A are consistent and the sFP_A–sFP_B relationship has been successfully discovered.

DA component has interfaces provided in Table 8-7 and DA component architecture is illustrated in Figure 8-9.

Table 8-7 – Discovery agent component interfaces

Input interface	Basic input parameters	Basic return parameters
Coordination in	Receives DA response message (responder DA identifier and pair of sFP identifiers)	
DM received	Initiator sFP identifier and initiator DA identifier	

Output interface	Basic output parameters	Basic return parameters
Coordination out	Sends DA response message (responder DA identifier and pair of sFP identifiers)	
DM sent	Initiator sFP identifier and initiator DA identifier	

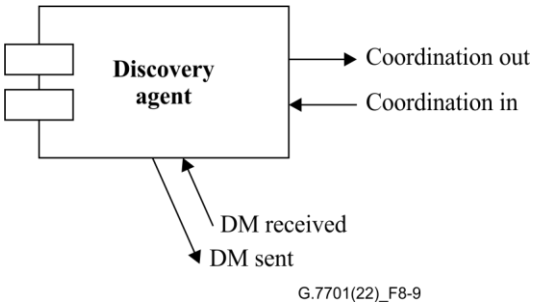


Figure 8-9 – Discovery agent component

Coordination in: The coordination in interface receives the DA discovery response message that includes the responder DA identifier and the pair of sFP identifiers.

DM received: This interface receives the in-band discovery message that was sent by the initiator sFP that includes the initiator sFP identifier and the initiator DA identifier.

Coordination out: The coordination out interface sends a discovery response message including the responder DA identifier and pair of sFP identifiers.

DM sent: The initiator DA uses this interface to trigger the initiator sFP to inject an in-band discovery message that contains the initiator DA identifier and the initiator sFP identifier.

Specifics of the discovery message are dependent on specific technology but there are no differences in the discovery procedure for circuit-switched networks, connection oriented packet switched networks and connectionless oriented packet switched networks.

8.3.6 TAP component

8.3.6.1 Termination and adaptation performers

The TAP is collocated with the resources (i.e., the adaptation and termination functions). It provides the LRM with a view of the resources that a support a link end; it abstracts the hardware and technology with specific details of the adaptation and termination functions. When utilized in an SDN control context, the TAP component resides with SDN controllers that can directly configure forwarding in the resources associated with the controller. TAP holds the relationship between SNP and FP name spaces. The identifiers in the FP name space are referred to as "labels" in this clause.

8.3.6.2 TAP resource model

Only those transport resources that are allocated to an MC component are made visible to the TAP. When a resource is permanently withdrawn (unallocated) from a control function, the SNP that references that resource should be deleted. 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).

The relationship between the TAP, LRM and the network resources is shown in Figure 8-10.

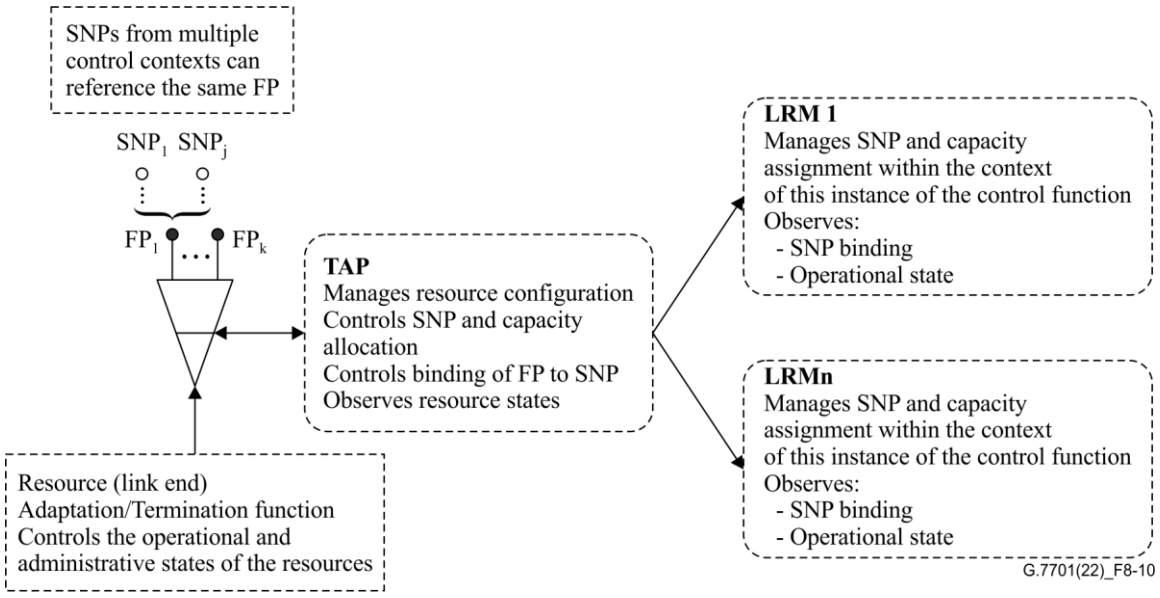


Figure 8-10 – Relationship between TAP, LRM, and transport resources

NOTE – The multiple control contexts may exist because of different VNs or different levels in a controller hierarchy, or both.

The TAP operates at two different times and provides two different functions.

When a resource is assigned to a control function, the TAP is configured with a list of the resource identifiers, the capacity of the link resource, together with the capacity reservation policy. The link resources may be shared between multiple control functions (e.g., different layer networks or different VNs, see clause 7.4). 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 SNPs and capacity to each LRM. In a circuit-switched network with a fixed bitrate, only the resource labels are configured since they carry an implicit resource capacity and reservation policy. In the case of a time division multiplexing (TDM)

network with flexible bitrate (e.g., OTN with ODUflex) the TAP provides the potential and allocated capacity to each LRM, together with the capacity assignment policy. In the case of packet-switched networks, the TAP provides the potential and allocated capacity (CIR and EIR) to each LRM, together with the capacity assignment policy. The LRM can only assign SNP link connections (to a connection) if the SNPs and resource capacity has been allocated by the TAP.

The TAP makes resources visible to an LRM by associating 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 with fixed capacity, when the TAP allocates the resource to an LRM (i.e., sets the SNP state to allocated), it also configures those resources and creates the FP and the link connection is created. This configuration action is independent of the assignment of those resources to a connection.

In the case of a circuit-switched network with flexible capacity, when the TAP allocates capacity and SNPs to a LRM (i.e., sets the SNP state to allocated), it only performs the configuration required to allow those resources to be activated. When the LRM assigns an SNP to a connection, the TAP creates the FP (i.e., it activates the binding between the SNP and the FP), it also configures the resource to support the requested capacity.

In the case of a packet-switched network, when the TAP allocates capacity and SNPs to an LRM (i.e., sets the SNP 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 FP (i.e., it activates the binding between the SNP and the resource label), it also configures the traffic conditioning and traffic policing functions, if required.

Figure 8-11 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. For a given label in the transport resource name space, the pair of SNP identifiers allocated for it are the same on both ends of the link. That is, $S_{nA}=S_{nZ}$, for n=1..m in Figure 8-11.

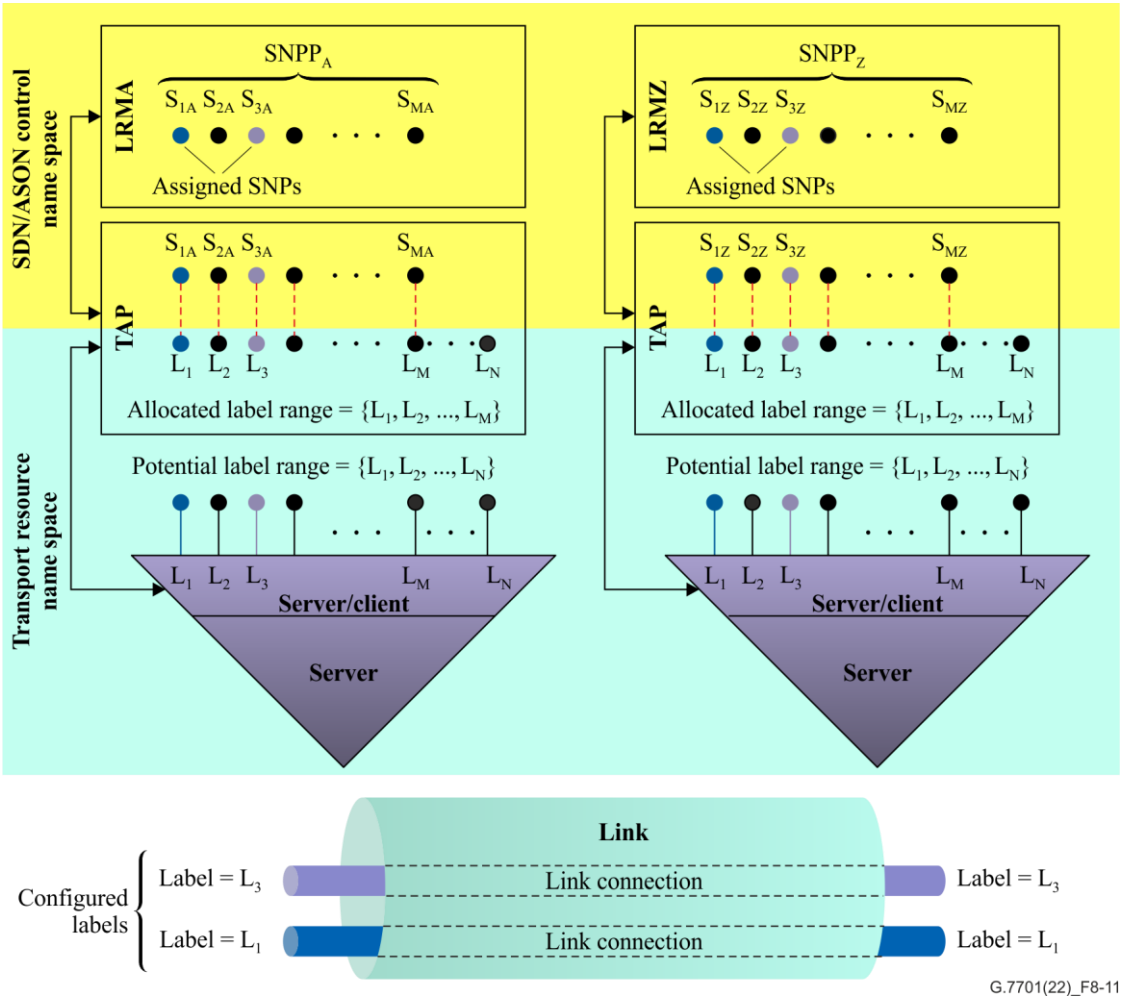


Figure 8-11 – MC components and link resource model

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 resource 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} = 1048576 possible label values including reserved labels (label values 0..15) for specific purposes.

– **Configured (resource) label**

A "configured label" is a transport resource label that has been configured 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 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. 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 connection request.

The allocated labels are entities that can be referenced in the transport resource name space. Each allocated label is associated with one or multiple SNP identifiers that exist in the MC component name space (1:n relationship). In the simplest case, there is exactly one SNP identifier per allocated label (1:1 relationship between allocated label and SNP identifier). 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 8-12 depicts how VNs can be modelled. In the provided example, the allocated label range is subdivided into three subsets, a subset that can exclusively be used by VN1, a subset that can exclusively be used by VN2, and a subset that is shared between VN1 and VN2. Each label of the subsets for exclusive use has a single SNP associated, whereas each label of the shared subset is associated with two SNPs, one SNP within the scope of LRM_{VN1} and another one SNP within the scope of LRM_{VN2}. When an SNP is assigned, e.g., by LRM_{VN1} that corresponds to a label from the shared label subset, the SNP in LRM_{VN2} becomes busy.

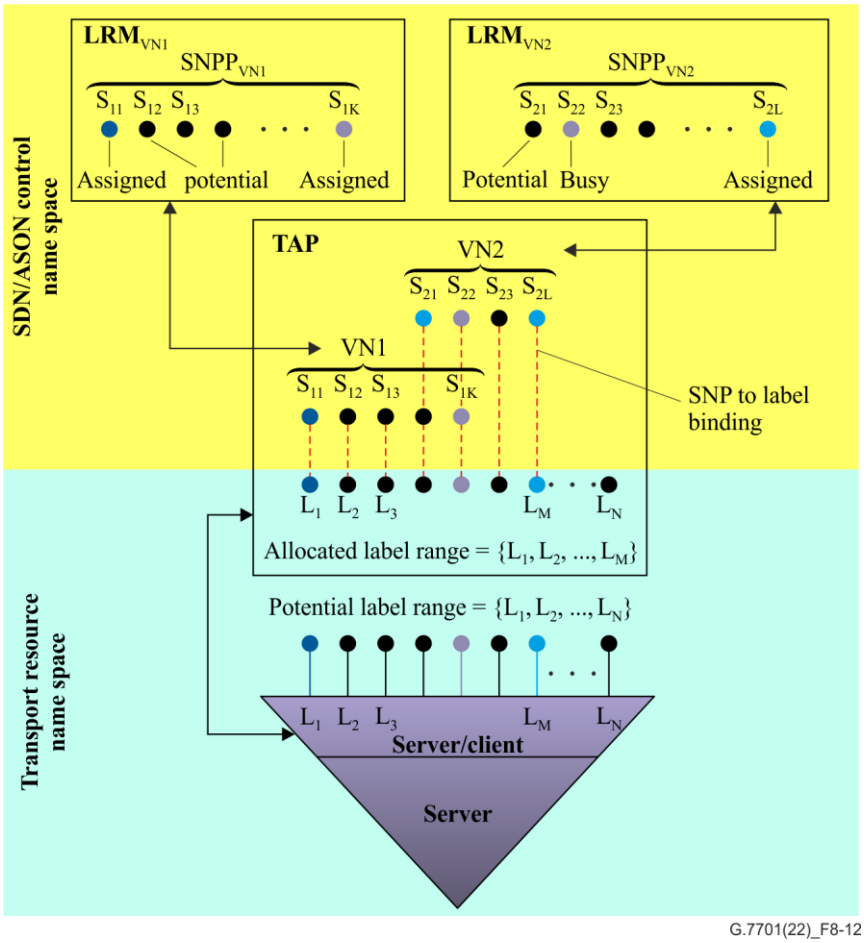


Figure 8-12 – MC component organizational model for VNs

8.3.6.3 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 8-8, the SNP binding states that the TAP provides to the LRM are constrained by the administrative state of the resources.

The transport 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 8-8 – SNP binding states

State	Description
Busy	Permitted binding, the resource label and capacity being referenced by the SNP is currently allocated to another control function or the management function.
Potential	Permitted binding, currently the resource label and capacity being referenced by the SNP is not allocated to any control function or the management function.
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 FP, the TAP is responsible for holding the SNP-FP binding. A local TAP cooperates with a remote TAP via the LRM to coordinate any variable adaptation or other coordination required when forming the FP 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 setup, a pair of TAPs cooperate via the LRM to coordinate any adaptation setup, 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 from the adaptation and termination functions to ensure that the management function indications are consistent. Management function 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:

1) 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.

2) administrative: This state reflects the permission to use the resource which is managed by a management interface to the TAP.

3) usage: This state reflects whether the resource is actively in use. As TAP allocates resources to and withdraws resources from the control function, it adjusts the usage state accordingly.

Permitted combinations of the resource states and the SNP binding state for each SNP are described in Table 8-9 below:

Table 8-9 – 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 RC 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 function or when the resource is being withdrawn from the control function. The management function will operate directly on the transport function resources. Changes to the [ITU-T X.731] states will not be visible to the TAP during this time.

8.3.6.4 Adding/removing resources from a control function

The resource administrative and usage states may be used to control the addition or withdrawal of a resource from the control function. This is illustrated in the administrative state transition diagram in Figure 8-13.

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

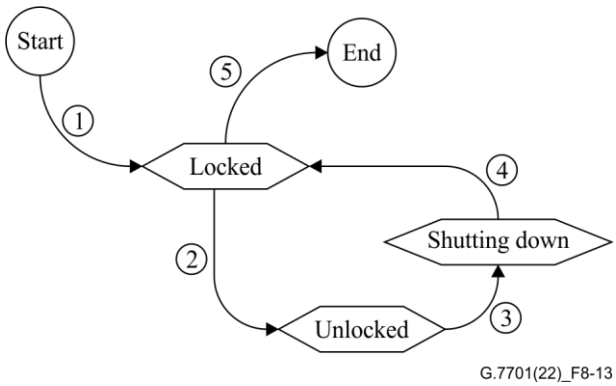


Figure 8-13 – Resource administrative state

Table 8-10 – Transition descriptions

Transition	Description	Interface
1	Resource is made visible to the TAP	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 function: If the withdrawal is permanent, then the TAP is instructed to delete all SNPs that referenced the resource	Resource state

8.3.6.5 SNPx binding state transitions per LRM

Figure 8-14 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.

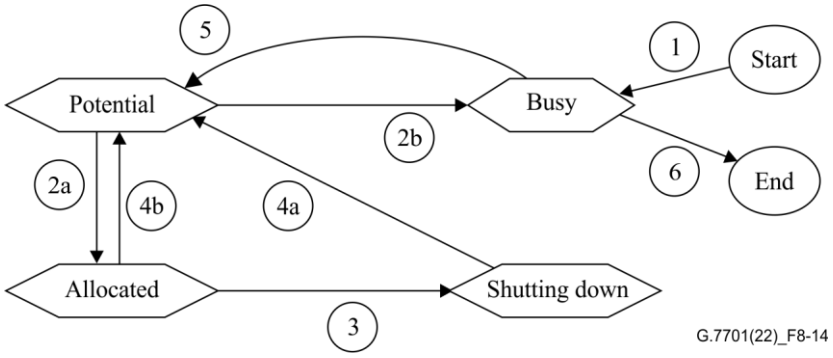


Figure 8-14 – SNP binding state in LRM

Table 8-11 – 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
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

8.3.6.6 TAP component interfaces

TAP component has the interfaces provided in Table 8-12 and the component is illustrated in Figure 8-15.

Table 8-12 – TAP component interfaces

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
Resource Configuration in	Resource configuration information FEF information;	Confirm
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
Fault reporting in	List of FP ids, Faults (reason, status)	Hardware specific

Output interface	Basic output parameters	Basic return parameters
Control	Hardware specific	Hardware specific
Resource configuration out	List of SNP ids CIR and EIR, capacity assignment policy (packet switched only)	confirm
Capacity change (packet switched only)	CIR EIR	Confirm

Table 8-12 – TAP component interfaces

Output interface	Basic output parameters	Basic return parameters
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 adaptation and termination functions.

Administrative state: This interface accepts administrative state.

Resource configuration in: This interface receives the request from LRM to configure the link end. This interface can be used to receive the FEF information from LRM for the SNP that is indicated to use it.

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.

Fault reporting in: This interface is used to accept fault information from transport resource.

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

Resource configuration out: 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 adaptation and termination functions.

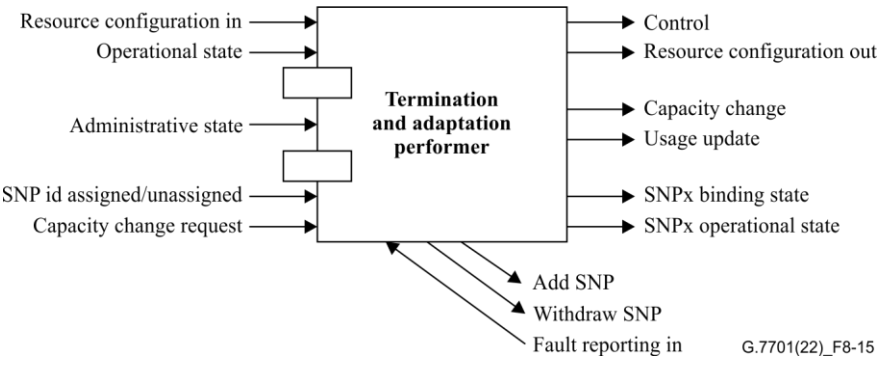


Figure 8-15 – Termination and adaptation performer component

8.3.7 Directory service component

The directory service component provides translation between identifiers in different name spaces and coordination among peer directory service components. Figure 8-16 shows the directory service component.

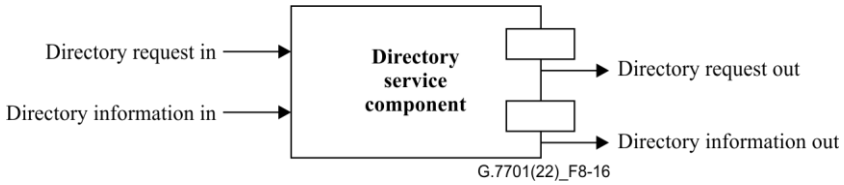


Figure 8-16 – Directory service component

The input and output interfaces for the DS are described in Table 8-13:

Table 8-13 – Directory service (DS) component interfaces

Input interface	Basic input parameters	Basic return parameters
Directory request in	1) Boundary resource identifier (BRI); or 2) Boundary resource identifier (BRI) alias; or 3) SNPP identifier; or 4) SNPP alias	1) Boundary resource identifier (BRI); or 2) Boundary resource identifier (BRI) alias; or 3) SNPP identifier; or 4) SNPP alias
Directory information in	1) <Boundary resource identifier (BRI), SNPP identifier> 2) <Boundary resource identifier (BRI) alias, Boundary resource identifier (BRI)> 3) <SNPP identifier, Boundary resource identifier (BRI)> 4) <SNPP alias, SNPP identifier> 5) <SNPP identifier, SNPP alias> 6) list of Boundary resource identifier (BRI)	

Table 8-13 – Directory service (DS) component interfaces

Output interface	Basic output parameters	Basic return parameters
Directory request out	1) Boundary resource identifier (BRI); or 2) Boundary resource identifier (BRI) alias; or 3) SNPP identifier; or 4) SNPP alias	1) SNPP identifier; or 2) Boundary resource identifier (BRI); or 3) Boundary resource identifier (BRI); or 4) SNPP identifier.
Directory information out	1) <Boundary resource identifier (BRI), SNPP identifier> 2) <Boundary resource identifier (BRI) alias, Boundary resource identifier (BRI)> 3) <SNPP identifier, Boundary resource identifier (BRI)> 4) <SNPP alias, SNPP identifier> 5) <SNPP identifier, SNPP alias> 6) list of Boundary resource identifier (BRI)s	

8.3.8 Notification component

The notification component receives the notifications from other local MC components (i.e., MC components) in the same MC system using the notification in interface. After processing, the notification component delivers these notifications to one or more external clients via the protocol controller component using the notification out interface. The notification component maintains a historical log of the notifications that have been sent to a client. The client may retrieve the history log using the notification query interface. The processing of notifications may include alarm correlation and alarm filtering. The notification component also manages subscriptions for notifications. The notification component provides the interfaces given in Table 8-14. The notification component is illustrated in Figure 8-17.

Table 8-14 – Notification component interfaces

Input interface	Basic input parameters	Basic return parameters
Notification subscription request in	Subscription filter	Notification subscription service
Notification query	Query filter	Notifications
Notifications in	Notifications	-

Output interface	Basic output parameters	Basic return parameters
Connection correlation query	Original resource identifiers	Relevant state
Notifications out	Processed notifications	-

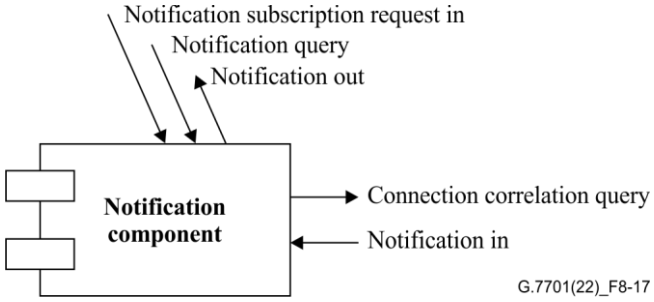


Figure 8-17 – Notification component

Notification subscription request in: This interface is used to receive a notification subscription request from a client of the MC system.

Notification query: This interface is used to retrieve the history of the notifications that have been sent to a client.

Notifications in: This interface is used to receive notifications from other MC components in the same MC system.

Connection correlation query: This interface is used to query the CC to get the relationship between the resource that generated the notification and the connections which are supported by that resource.

Notifications out: This interface is used to deliver notifications to external clients.

8.3.8.1 Alarm notification

Alarms are a type of notification (clause 8.3.8). One of the primary capabilities of alarm management is alarm correlation.

Alarm correlation refers to the analysis of the relationship between the resource that generates alarms and the (virtual) resources occupied by connections handled by the MC system, to determine whether the alarm has an impact on the connections managed by the MC system.

8.3.8.2 Notification subscription

Subscription is a mechanism that allows a client to receive notifications about specific parameters in the network provided by the MC system (i.e., the server). The client and the server need a common set of criteria on the establishment, maintenance, history retrieval, modification, and deletion of a subscription. A batch of notifications may be sent to the client at specific times, or a notification may be sent immediately for each event. This behaviour is defined in the subscription.

The notification subscription is processed by the notification component, the notification subscription request in interface is used to receive the request to establish, modify or delete the notification subscription, and the feedback is the notification subscription service.

8.3.9 Protocol controller (PC) component

The protocol controller maps the parameters from one or more interfaces of one or more MC components into messages that are carried by a protocol across the CCN. It also demultiplexes messages received from the CCN into parameters that are passed to other MC components. The protocol controller is attached to the CCN via a reference point (for example a CPI, UNI, I-NNI, E-NNI). The protocol controller provides authenticated, secure, and reliable transfer of parameters across the CCN. This permits transactions (e.g., between MC components) to be tracked to ensure expected responses are received, or that an exception is reported to the originator of a request. The protocol controller provides a PEP (see clause 8.2), the protocol controller reports security violations (i.e., messages that do not conform to the PEP policies) via its notification interface.

The protocol controller MC component maps the parameters from the interfaces of the MC components in the same MC system into (external) messages that are carried by a protocol to support interconnection via an external interface. The mapping process is transparent i.e., it is independent of the semantics of the parameters being mapped. The protocol controller may also multiplex several MC component interfaces into a single instance of a protocol implementation as shown in Figure 8-19b. In this case the protocol implementation supports multiple abstract interfaces.

The protocol controller component provides the interfaces given in Table 8-15. The protocol controller component is illustrated in Figure 8-18.

Table 8-15 – Protocol controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Local parameters in	Parameters from other MC components in the same MC system	-
Notifications in	Notifications from a local notification component	-
Message in	Encoded parameters	Acknowledgement to peer protocol controller

Output interface	Basic output parameters	Basic return parameters
Message out	Encoded parameters	Acknowledgement from peer protocol controller
Remote parameters	Parameters extracted from the Message in interface	-
Notification out	Protocol exceptions	-

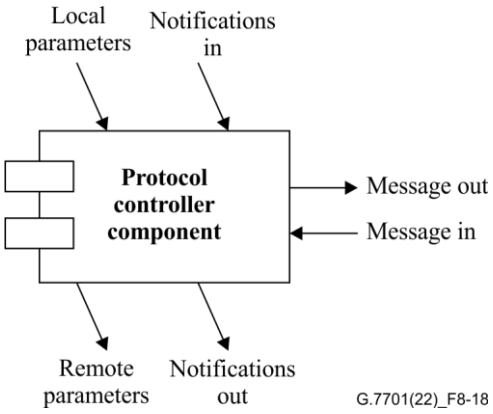


Figure 8-18 – Protocol controller component

Local parameters in: Parameters from the interfaces of local MC components that need to be communicated to an external client

Notifications in: Notifications from a local notification component

Message in: Messages carrying encoded parameters from a peer protocol controller component.

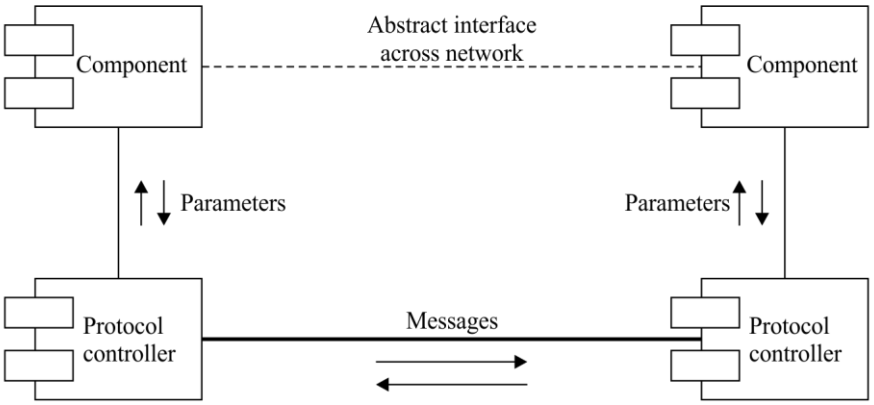
Message out: Messages to a peer protocol controller component that carry encoded parameters from local MC components or notifications from a local notification component.

Remote parameters: Parameters extracted from messages received from a peer protocol controller component. These parameters are pass to a local MC component. The parameters are only provided on this interface if message conforms to the PEP policies. If the expected response to a request is not received an exception is returned to the originator of the request.

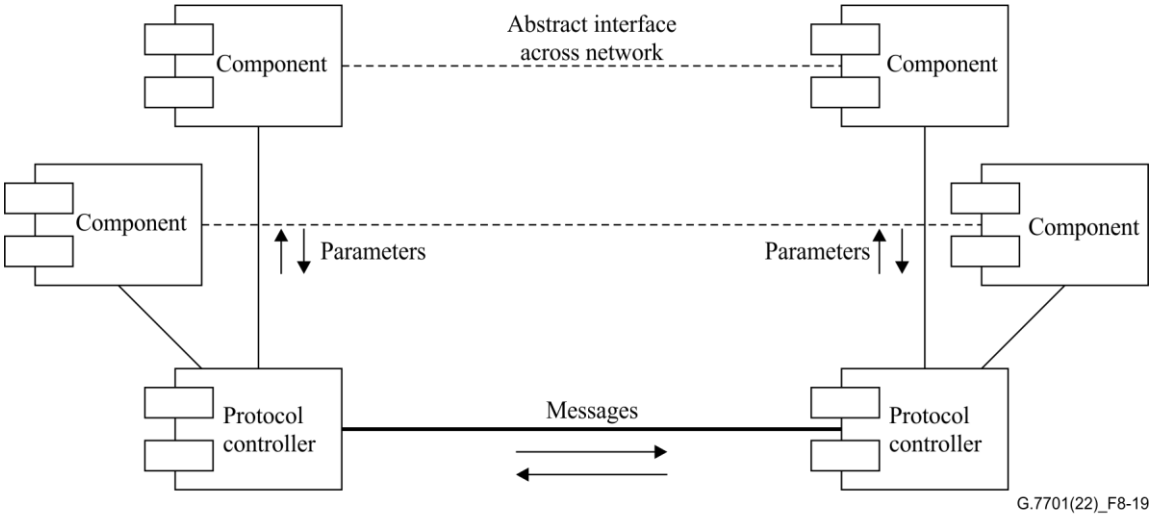
Notification out: Protocol exceptions including security violation (generated as the result of receiving a message that does not conform to the PEP policies) or session status (e.g., unacknowledged messages).

The details of an individual protocol controller are dependent on the protocol being used.

Parameters are passed between MC components in different MC systems via the protocol controller. The protocol controller is semantically transparent to parameters being passed. The parameters are mapped into external protocol messages and vice versa. Messages are passed between the two protocol controllers. This is illustrated in Figure 8-19a for the case where a protocol implementation supports a single abstract interface between the MC components. Figure 8-19b illustrates the case where a protocol implementation supports two abstract interfaces between the MC components.



a) Generic use of a protocol controller



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

Figure 8-19 – Protocol controller

8.3.10 Traffic policing (TP) component

This component acts in the FP name space to check that the incoming user connection is sending traffic according to the parameters defined by the call. If the user traffic violates the agreed parameters, then the TP may instigate measures to correct the situation. The TP policies may be requested in the call/connection set-up process.

NOTE – This is not needed for a continuous bit rate transport layer network and is not further expanded in this Recommendation.

9 Common control communications

Various applications (management, SDN, ASON, overhead communications network (OCN), etc.) require a communications network to transport information between a variety of components, as illustrated in Figure 9-1. [ITU-T G.7712] specifies DCN functions that can be used to support the

communications requirements of one or more application (e.g., the CCN to that supports communications between MC systems.).

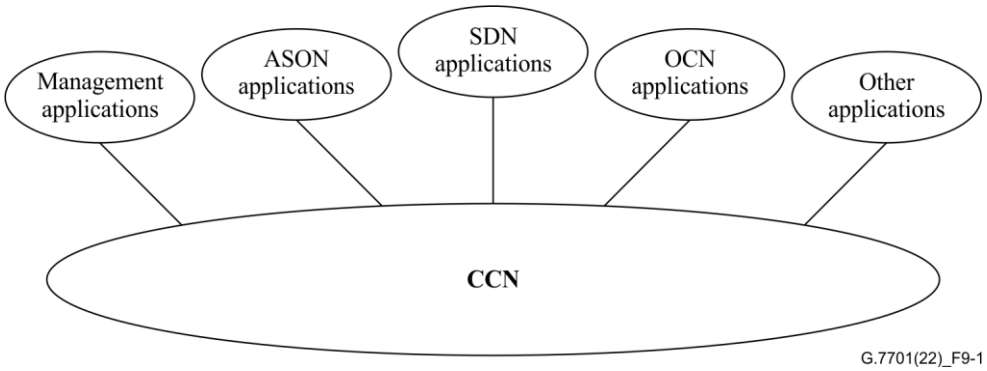


Figure 9-1 – Example applications supported by a CCN

10 Common management aspects of MC components

This clause introduces constructs relevant to the management of MC components.

10.1 MC components

The following management functions apply to the MC components.

– TAPs require configuration management.
– DAs require fault management, configuration management.
– LRMs require configuration management, and performance management.
– NCCs require performance management including call statistics, e.g., number of call completed, number rejected, etc. NCCs also require configuration management.
– RCs require configuration management.
– CCs require configuration management, and performance management.
– DSs require configuration management.
– Notification components require configuration management.
– PCs require fault management and configuration management.

10.2 Control function management requirements

The requirements for the management of the control function management are provided in [ITU-T G.7718].

11 Identifiers

Distinct and independent sets of name spaces exist, from which identifiers are drawn, for:

– resources in the transport network;
– control view of transport resources;
– MC components;
– control artefacts;
– reference points;
– control communications network.

11.1 Resources in the transport network

The architecture of transport resources is described in [ITU-T G.800] and summarized in clause 7.1. The identifiers used by transport resources to allow it to deliver communications from a source to a defined set of destinations are described in [ITU-T G.800] (clause 7, Annex A.4 and Appendix I). [ITU-T G.800] describes topological components that are relationships between reference points, for example, the subnetwork and link. Transport processing functions in the architecture manipulate information at points, and include functions such as adaptation and termination. Transport entities provide the means to transfer information and include connections.

A topological component may have zero or more identifiers associated with it. Similarly, transport processing functions and transport entities may have zero or more identifiers associated with it. A common use of multiple identifiers for the same component is when multiple applications refer to the same component.

This Recommendation assumes that there is a singular [ITU-T G.800] FP name space for a set of resources in a layer network. The identifiers drawn from this name space are used by the transport resources to allow the delivery of communications from a source to a sink. Examples of these resource identifiers are; MPLS label; Ethernet SA and DA; wavelength; the TS of an ODU server. The only MC components that use identifiers from the FP name space are the DA and TAP.

11.2 Control view of transport resources

Name spaces need to be understood by an MC system. The directory service (DS) component provides a translation function to convert the name space used by one MC system to the name space used by another MC system. It is also possible to have name space translation between MC components within an MC system.

For example, as described in clause 8.3.1, the NCC can request DS to translate a boundary resource identifier (BRI) into an SNP/SNPP identifier.

The MC components use three different name spaces to reference the transport resources:

– Routing area name space and subnetwork name space provide identifiers for [ITU-T G.800] topological entities (subnetworks and links).

– Link context name space that provides the identifiers (SNPs) for the ends of transport entities ([ITU-T G.800] FPs).

– FP name space used by the TAP and DA to directly configure forwarding in transport entities. The TAP and DA provide a mapping between the SNP identifier and the forwarding resource identifier[1].

Independent identifiers (drawn from these name spaces) may be used for each layer network, client control domain and server control domain. Normally the routing area identifiers and link identifiers (SNPPs) are structured in a way that simplifies the implementation of routing and connection management. For example, in the case of hierarchically arranged routing areas it may be convenient to use recursive (hierarchical) identifiers for the contained SNPP links or routing areas.

11.2.1 Name spaces for routing and connection control

MC components that perform routing control operate on a particular set of transport resources within the scope of their designated routing domain/routing area. As defined in clause 3.2, 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. Thus, MC components collectively use three

[1] In some cases, in an implementation it may be convenient to reuse the value of the forwarding resource identifier as the SNP identifier at higher levels in the controller hierarchy. However, in this context the resource semantics are absent, and the value is treated as an opaque SNP identifier.

separate name spaces in order to reference routing area and SNP/SNPP abstractions. These are the routing area name space, subnetwork name space, and link context name space.

The link context name space specifies within the SNPP where the SNP is. At some (low) level in a hierarchy of controllers the SNP identifier must be mapped to a resource label and the TAP component can maintain this mapping. The TAP makes resources visible to an LRM by binding a transport resource label to an SNP identifier.

An SNPP identifier is a concatenation of:

– one or more nested routing area identifiers;

– an optional subnetwork identifier within the lowest routing area level; this can only exist if the containing RA identifiers are present;

– one or more nested resource context identifiers.

This Recommendation does not specify formats or values for identifiers in instances of the aforementioned name spaces. It is possible for a network implementation to assign the same value to identifiers in different name spaces (e.g., SNPP identifier and BRI) in order to simplify configuration.

11.2.2 Name space recursion

An SNPP identifier can recur with routing areas as the routing area name space is typically hierarchical. As recursion is a property of [ITU-T G.800] subnetworks it would be natural, though not mandatory, for the naming associated with subnetworks to also be hierarchical.

[ITU-T G.800] also describes recursion between layers using adaptation and termination functions. While name spaces are independent between layers, if an operator owns and manages a client layer supported by a server layer, it is a common operational requirement to understand how a server layer trail can support a client layer connection and, in this case, a common name space may be used.

Some relationship between recursive name spaces thus needs to be maintained and one mechanism for managing such name space interactions is through the use of the directory service component.

11.3 MC components

MC components also require separate name spaces, as they may be instantiated differently from each other for a given control function instance. For example, and implementation may have a centralized NCC with distributed CCs. Thus, separate identifiers are needed for RCs, NCCs, and CCs.

Additionally, the PCs that are used for protocol specific communication also require a separate name space and this is referred to as the CCN name space (see clause 11.6). The separate name spaces ensure that there is no dependency on the identifiers of the resources that the MC component have in scope (e.g., routing area), the identifier of the MC component itself, or the CCN address that is used to deliver messages to that MC component. This independence allows, for example, the location or scope of an MC component to be changed without modifying its identifier.

The MC component name space is also used for configuration and fault reporting of the MC components.

11.4 Control artefacts

Control functions create and use artefacts including, for example, connections, routes, calls, and directories. Normally the MC component that creates a control artefact assigns an identifier. These artefacts have identifiers associated with them and are drawn from an independent name space.

11.5 Reference points

A reference point represents a collection of services, provided via interfaces on one or more pairs of MC components. The MC 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 MC components supporting the interface are not visible, hence the interface specification can be treated independently of the MC components.

Various logic reference points are defined, for example the UNI, I-NNI and E-NNI reference points in [ITU-T G.7703] and the CPIs in [ITU-T G.7702].

At the boundary of administrative domains, the BRI is used to identify both the transport resources and the interface between MC systems that are exposed at the domain boundary. The BRI is drawn from an independent name space. The use of an independent name space avoids exposing the identifiers that the MC components use within the domain.

11.6 Control communications network

To enable MC components to communicate with each other, a control communications network (CCN) is used as specified in [ITU-T G.7712]. CCN addresses identify the points of attachment for the PCs that instantiate control communication functions (generating and processing messages in protocol specific formats). Each PC has a distinct CCN point of attachment (CCN attachment).

MC components access the CCN via a PC, a PC may support one or more MC components. A directory may be used to relate the MC component identifier with the CCN address of its (current) PC. The independence between the MC component identifier and the CCN address allows, for example, the location of an MC component to be changed, or for the CCN to be reconfigured, without modifying the identifier of the MC component.

12 Resilience

Resilience refers to the ability of the MC components to continue operating under failure conditions. Operation of the MC components depends upon elements of the control communications network (CCN), the transport network, and the components of the MC system itself.

12.1 Principles of MC component and transport network interactions

The following principles are used for MC component and transport network interactions when communications become available between them.

1) The MC component relies on the transport network for information about transport resources.
2) Consistency between the control view and the corresponding transport resources is established first.
3) MC components synchronize with their adjacent MC components. This is used to re-establish a consistent view of routing, call, and connection state.

Another principle of MC component and transport network interaction is that:

4) existing connections in the transport network are not altered if the MC component fails and/or recovers.

For resiliency, the transport network resource and SNC state information should be maintained in non-volatile store. Further, some information about the MC component use of the SNC should be stored. This includes whether the SNC was created by CC and how it was used. For example, which end of the SNC is towards the head end of the whole connection. The MC components must ensure it has resource and SNC state information that is consistent with the resource and SNC state information maintained by the transport network. If not, the MC components must:

- ensure that 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 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 MC components reconstruct the call and connection state corresponding to existing connections. For example, RC will need to disseminate correct SNP information after it is synchronized with LRM.

The MC components re-establishment of information consistency with the transport node should occur in the following sequence:

- the LRM synchronizes the transport resource state information via the TAP;
- the CC then synchronizes with the LRM;
- the NCC then synchronizes with the CC.

Following the re-establishment of state consistency, the MC components must then ensure SNC state information consistency with adjacent MC components, as discussed in principle 3 above, prior to participating in connection set-up, modification 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 network operator may be notified if the failure persists and requires operator intervention (for example, to release a call).

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

13 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 MC components provide a network operator with the ability to offer a user call 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 (none, protection, restoration or both) 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 multiple domains (e.g., control or recovery domains), each domain applies its availability enhancement techniques to the segment(s) of the call/connection(s) it supports.

The protection or restoration of a connection may be invoked or temporarily disabled by a command from network operator. 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 of client layer connections that need to be restored simultaneously and rapidly.

– utilize a robust control communication mechanism, which remains functional even after a failure in the transport network or CCN.

– 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.

13.1 Protection

Protection is a mechanism for enhancing the 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 main difference between protection and restoration is that in the event of a failure, protection does not involve re-routing or additional connection set-up.

The CC component is responsible for the creation of the connections. This includes creating both a working connection and a protection connection including the configuration information for the selected protection scheme.

The selection of the working or protection connection can be performed by either: The transport resources, or; the CC responsible for the connection in the associated protection domain.

13.2 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. Restoration occurs in relation to re-routing domains (see clause 7.2) or recovery domains. A re-routing domain is a group of call and connection controllers that share control of domain-based re-routing. MC components in 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 domain or area. A routing 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 entirely 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 by 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, MC components of each re-routing domain determine 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 are not changed. Requests for inter-domain

re-routing service 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.

During the negotiation of the re-routing services, MC 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 all re-routing domains involved.

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, an alternative connection segment is created between the edges of the re-routing domain. This alternative connection is 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 13-1. In this example, the routing domain is associated with a single routing area and a single re-routing domain. The call is re-routed by the associated MC components within the re-routing domain.

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 by the network operator, the MC components establish a re-routing connection. Once the re-routing connection is created, the CC 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 RC 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), NCCs must not release the original (failed) connections. The NCCs must continue monitoring the original connections, and when the failure is repaired, the call is restored to the original connections.

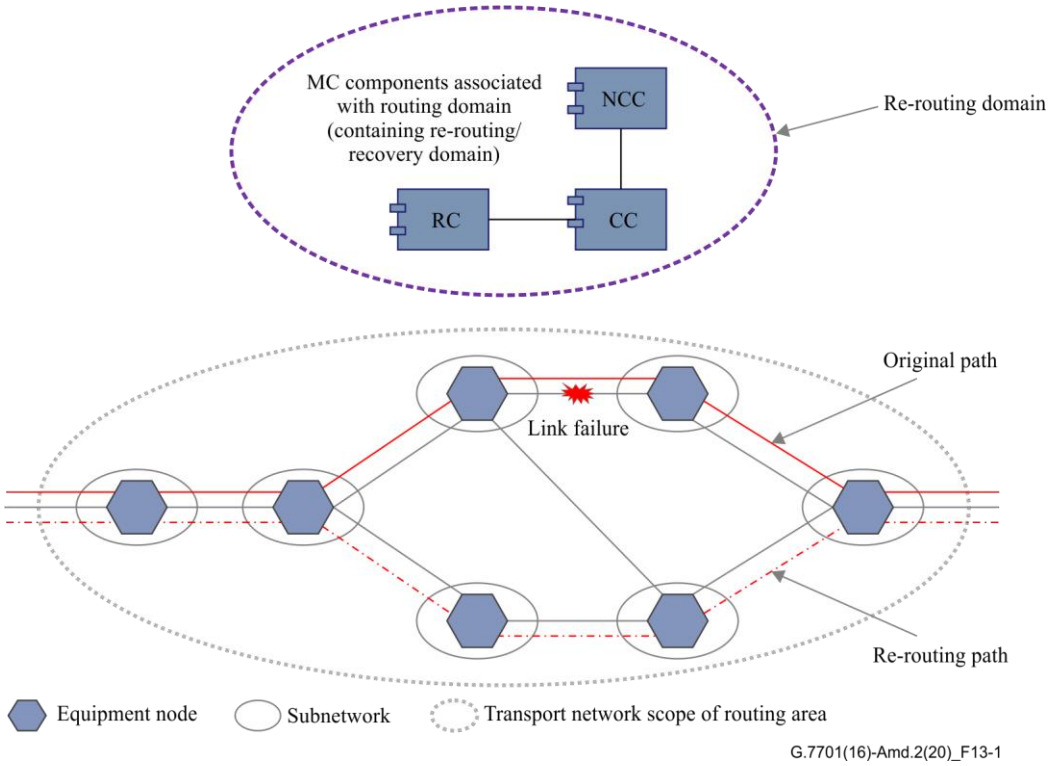


Figure 13-1 – Example of hard re-routing

13.2.1 Re-routing in response to failure

13.2.1.1 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.

13.2.1.2 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.

13.2.1.2.1 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 13-2-a) no re-routing operation can be performed. In this case, alternative protection mechanisms may be employed between the domains.

Figure 13-2-b shows the example with two links between domain A and domain B. A link between domains with the appropriate level of protection must be selected by RC component. 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.

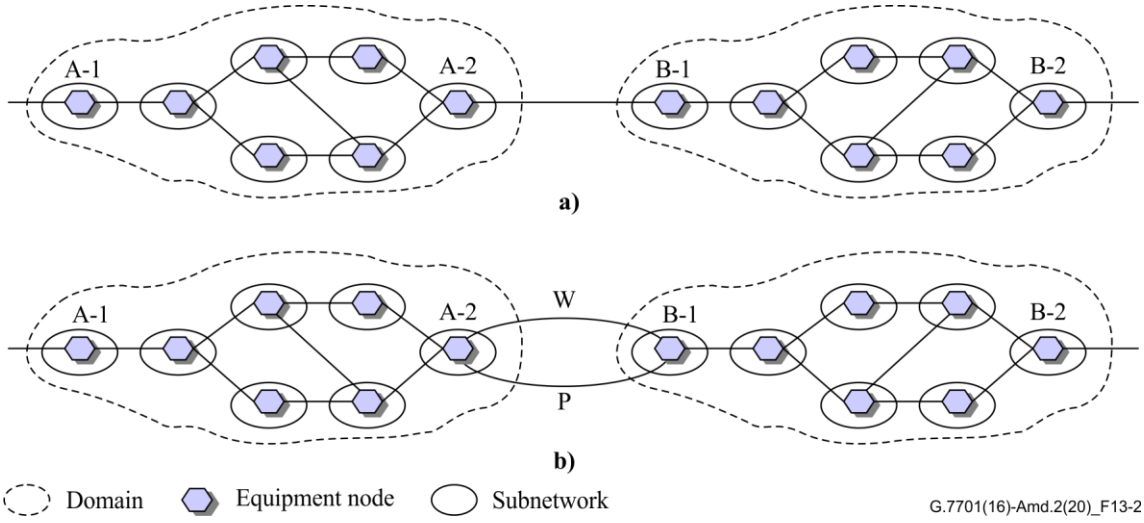


Figure 13-2 – Link failure scenarios

13.2.1.2.2 Gateway network element failure

This case is shown in Figure 13-3. 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 A2) a new connection should be created to support the call. 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.

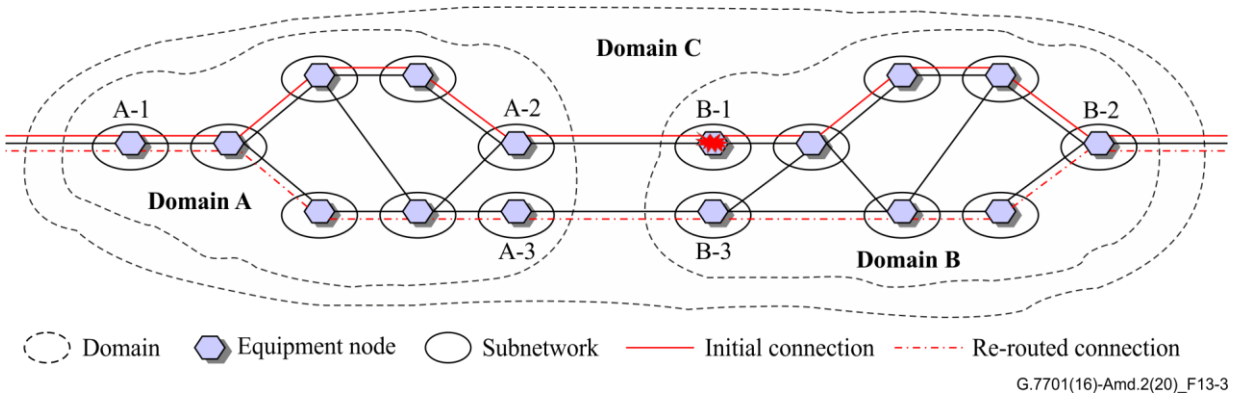


Figure 13-3 – Re-routing in event of a gateway network element failure

13.3 Nested routing domains

Protection, re-routing and recovery domains are types of routing 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.

Annex A

Configuration of OTN digital and media layers

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

[ITU-T G.872] describes the OTN digital layer networks and the media layer. The abstractions used to represent the digital layer networks and the media are similar. However, configuration of the media and the association of media channels to optical signals requires some special consideration.

The configuration of a digital layer network may be simplified since the hypothetical reference model used to allocate impairments has sufficient scope (and implementations have sufficient operating margins) that impairments do not need to be considered. Further, layer independence allows a client (digital) layer network to be configured in a way that is largely independent of the server (digital) layer network.

This simplification cannot be applied to all cases of configuring the media and optical tributary signal (OTSi). As a result, further requirements are imposed when configuring the media layer and the signals that cross that media layer.

– The media can be configured in advance of the assignment of the media channel to one or more OTSi
  - The configuration of a media channel is independent of the presence (or absence) of any optical signal.
– The width of the media channels may be different from the width required to carry a single OTSi. (e.g., a "wide" media channel may be used to provide what is commonly called an "express" channel in a ROADM)
– A single client digital stream (optical transport unit (OTU)) may be carried by more than one OTSi
  - Hence the configuration of the media to carry an OTU may require the configuration of one or more network media channels. Further, some of the optical tributary signal assembly (OTSiA) OAM functions are supported by non-associated overhead (OTSiG-O) and this must be configured when an OTSi is assigned to a network media channel.
– Determining the compatibility between an OTSi and a network media channel is a complex process. There is currently no standardized method defined for confirming compatibility in the case where the media is divided across dense wavelength division multiplexing (DWDM) line segments from more than one vendor as described in [ITU-T G.680] scenario 2. Three single domain cases must be considered:
  1) the black link approach as described in [ITU-T G.698.1] and [ITU-T G.698.2]. In this case, compatibility can be checked by considering the application codes defined in [ITU-T G.698.1] and [ITU-T G.698.2].
     – The optical domain is normally configured by vendor specific media control, which would not need to manage the optical signal terminations.
  2) pre-computed optical paths: In this case paths through the media network are pre-computed and only those that are compatible with the candidate OTSi are provided in response to request for a network media channel. In general, the determination of compatibility is vendor specific.
  3) compatibility check at the time of request: In general, the determination of compatibility is vendor specific. In this case paths through the media are computed, and after path computation is complete these paths are checked for compatibility with the OTSi.

Compatibility checking is a complex process that must take into account the interaction of the "new signal" with all existing or planned[2] signals that share the same fibre and the impact of these existing (or planned) signals on the candidate "new signal". The actual transfer characteristics of the media elements in the path (e.g., filters, amplifiers) also needs to be considered.

A media[3] control function must, in addition to the capabilities required to manage digital layer networks, provide the additional capabilities described below. Two scenarios should be considered.

1) In the first scenario the client requests a network media channel, where all network media channels, the optical source and sink are compliant with the black link approach as specified in [ITU-T G.698.1] and [ITU-T G.698.2]. The request must include the end points of the network media channel being requested and the application code. Specific details on the network media channel centre-frequency and spectral width may or may not be included in the request, depending on the control architecture of the implementation. The media controller must configure the non-associated overhead that is conveyed via the overhead communications channel (OCC) (defined in [ITU-T G.7712]), to support control/management of the network media channel. The client controller is responsible for ensuring that the OTSi provided across the interface is compliant with the network media channel requested and that the digital content of the optical signals passed across the network media channel can be successfully received at the far end. The media control function must also:
   - Support the configuration of the requested media channel and the non-associated overhead.
   - Support the configuration and monitoring of the optical multiplex section (OMS) and optical transmission section (OTS) maintenance entities.

   Where network media channels are not compliant with in [ITU-T G.698.1] or [ITU-T G.698.2], joint engineering is required to ensure proper interoperability between the client and media providers and their associated control functions (if separate). In this case, a connection request must include the end points of the network media channel being requested and a mutually agreed application identifier. Beyond that, the request contents are dependent on the decisions reached in the joint engineering of the network and control/management scheme employed.
2) In the second scenario the client network requests an OTU network connection. In this case the media control function must:
   - Accept requests for an OTU connection that is translated into an OTSiA connection.
     - This requires the capability to configure the media channels, the non-associated overhead and the optical termination source and sink. Some of the OTSiA OAM functions are supported by non-associated overhead and this must be configured when an OTSi is assigned to a network media channel.
     - As described above, ensure that the OTSi and network media channel are compatible.
     - Ensure that the OTU supported by the OTSi/OTSiA passed across the network media channel is delivered successfully to the destination interface.
   - Support the configuration of media channels to support the OTSiA.
   - Support configuration and monitoring of the OMS and OTS maintenance entities.

---

[2] For example, a fibre may only be carrying one other signal currently, but it is intended to support up to a total of 80 signals.

[3] The implementation of a media control function is vendor specific; it may be offered as a stand-alone application or integrated with the control function for digital layer networks.

Appendix I

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 (OTSiA) layer networks.

If the OTSiA 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 I.1 shows an example of an explicit multi-layer topology for such a network. In this example the transitions from the ODU layer to the OTSiA layer are shown using transitional SNPP links with an adaptation/termination icon. This indicates the transition from ODUk to OTUk to OTSiA occurs between the routing areas connected by these links. The structure of the OTSiA 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.

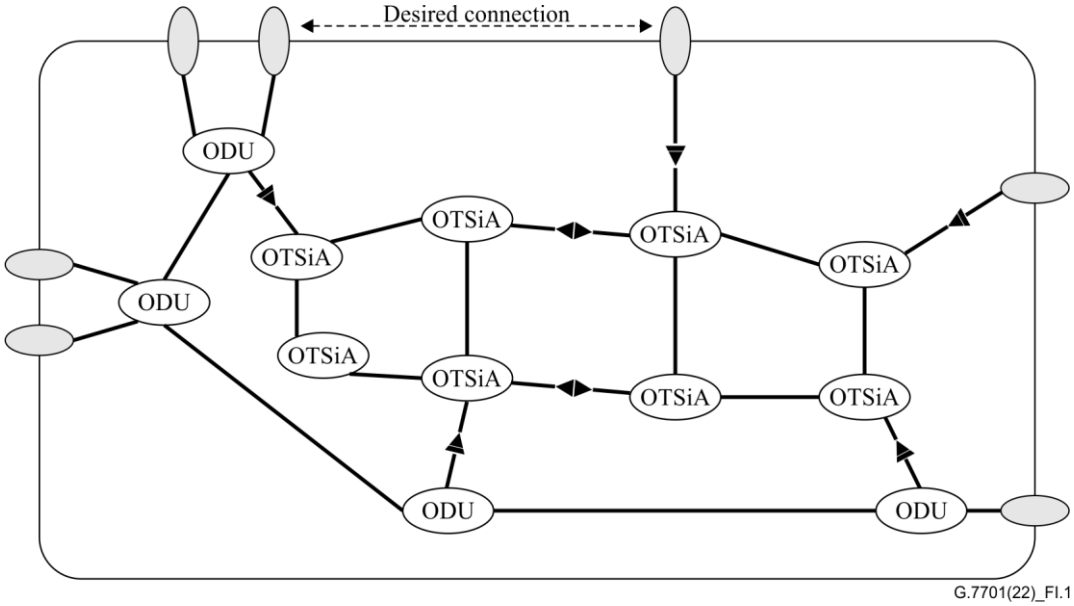


Figure I.1 – Explicit multi-layer topology example

Attributes can be associated with the transitional SNPP links and OTSiA layer SNPP links that are specific to OTSiA layer network routing (e.g., modulation type, FEC type, wavelength, etc.). These OTSiA 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 OTSiA routing areas and meet both ODU path constraints and, where necessary, OTSiA path constraints. This facilitates more optimal route selection across the entire network. The use of an explicit multi-layer topology in this case is particularly straightforward since the relationship between ODUk, OTUk, and OTSiA 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 M.3010] Recommendation ITU-T M.3010 (2000), *Principles for a telecommunications management network*.

[b-UML] OMG Unified Modeling Language® (UML®), Version 2.5.1
https://www.omg.org/spec/UML/

[b-ONF] Open Networking Foundation (2016), *SDN Architecture 1.1*, TR-521, February.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
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	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
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, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022