INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.8080/Y.1304

(11/2001)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

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

Digital networks – General aspects

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

Internet protocol aspects – Transport

Architecture for the automatically switched optical network (ASON)

ITU-T Recommendation G.8080/Y.1304

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	G.500–G.599
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DIGITAL TERMINAL EQUIPMENTS	G.7000–G.7999
DIGITAL NETWORKS	G.8000–G.8999
General aspects	**G.8000–G.8099**
Design objectives for digital networks	G.8100–G.8199
Quality and availability targets	G.8200–G.8299
Network capabilities and functions	G.8300–G.8399
SDH network characteristics	G.8400–G.8499
Management of transport network	G.8500–G.8599
SDH radio and satellite systems integration	G.8600–G.8699
Optical transport networks	G.8700–G.8799

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

ITU-T Recommendation G.8080/Y.1304

Architecture for the automatically switched optical network (ASON)

Summary

This Recommendation describes the reference architecture for the control plane of the Automatically Switched Optical Network that supports the requirements identified in ITU-T Rec. G.8070. This reference architecture is described in terms of the key functional components and the interactions between them.

Source

ITU-T Recommendation G.8080/Y.1304 was prepared by ITU-T Study Group 15 (2001-2004) and approved under the WTSA Resolution 1 procedure on 29 November 2001.

FOREWORD

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

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

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

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

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation G.8080/Y.1304

Architecture for the automatically switched optical network (ASON)

1 Scope

This Recommendation specifies the architecture and requirements for the automatic switched transport network as applicable to SDH transport networks, as defined in ITU-T Rec. G.803, and Optical Transport Networks, as defined in ITU-T Rec. G.872.

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

For the purposes of this Recommendation, components are used to represent abstract entities rather than instances of implementable software. UML-like notation is used to describe components of the architecture of the Automatically Switched Optical Network.

This is derived from the requirements defined in ITU-T Rec. G.8070.

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.

- ITU-T Recommendation G.705 (2000), *Characteristics of plesiochronous digital hierarchy (PDH) equipment functional blocks*.
- ITU-T Recommendation G.707/Y.1322 (2000), *Network node interface for the synchronous digital hierarchy (SDH)*.
- ITU-T Recommendation G.709/Y.1331 (2001), *Interfaces for the optical transport network (OTN)*.
- ITU-T Recommendation G.783 (2000), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks*.
- ITU-T Recommendation G.798 (2002), *Characteristics of optical transport network (OTN) hierarchy equipment functional blocks*.
- ITU-T Recommendation G.803 (2000), *Architecture of transport networks based on the synchronous digital hierarchy (SDH)*.
- ITU-T Recommendation G.805 (2000), *Generic functional architecture of transport networks*.
- ITU-T Recommendation G.807/Y.1302 (2001), *Requirements for the automatic switched transport network (ASTN)*.
- ITU-T Recommendation G.872 (2001), *Architecture of optical transport networks*.
- ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network*.
- ITU-T Recommendation M.3000 (2000), *Overview of TMN Recommendations*.

– ITU-T Recommendation M.3100 (1995), *Generic network information model*.

3 Definitions

This Recommendation defines the following terms:

3.1 Access Group (AG): See ITU-T Rec. G.805.

3.2 adaptation: See ITU-T Rec. G.805.

3.3 administrative domain: See ITU-T Rec. G.805.

3.4 agent: Within this Recommendation, the term agent is used to describe the entity that represents certain attributes and behaviour of a resource. The agent allows interaction between various resources and management and control functions. More than one agent may represent a resource.

3.5 call: An association between endpoints that supports an instance of a service.

3.6 component: An element that is a replaceable part of a system that conforms to and provides the realization of a set of interfaces.

3.7 connection: A connection is a concatenation of link connections and subnetwork connections (as described in ITU-T Rec. G.805) that allows the transport of user information between the ingress and egress points of a subnetwork.

3.8 Connection Point (CP): For the purposes of this Recommendation, a Connection Point represents the North input port of an Adaptation function. (Note that in ITU-T Rec. G.805 the CP refers to the binding between two points).

3.9 Connection Termination Point (CTP): See ITU-T Rec. M.3100, Amendment 1. A connection Termination point represents the signal state at the CP.

3.10 control plane: The Control Plane performs the call control and connection control functions. Through signalling, the control plane sets up and releases connections, and may restore a connection in case of a failure.

3.11 fabric: See ITU-T Rec. G.805.

3.12 link: See ITU-T Rec. G.805.

3.13 link connection: See ITU-T Rec. G.805.

3.14 management plane: The Management Plane performs management functions for the Transport Plane, the control plane and the system as a whole. It also provides coordination between all the planes. The following management functional areas identified in ITU-T Rec. M.3010 are performed in the management plane:

– performance management;
– fault management;
– configuration management;
– accounting management;
– security management

The TMN architecture is described in ITU-T Rec. M.3010, additional details of the management plane are provided by the M-series Recommendations.

3.15 policy: The set of rules applied to interfaces at the system boundary, which filter messages into an allowed set. Policy is implemented by "Port Controller" components.

3.16 port controller: A class of component that implements the set of rules applied to a system.

3.17 subnetwork: A topological component used to effect routing of a specific characteristic information. For the purposes of this Recommendation, a subnetwork is bounded by Subnetwork Points.

3.18 Subnetwork Connection (SNC): A subnetwork connection is a dynamic relation between two (or more in the case of broadcast connections) Subnetwork points at the boundary of the same subnetwork.

3.19 Subnetwork Point (SNP): The SNP is an abstraction that represents an actual or potential underlying CP (or CTP) or an actual or potential TCP (or TTP). Several SNPs (in different subnetwork partitions) may represent the same TCP or CP.

3.20 Subnetwork Point Pool (SNPP): A set of subnetwork points that are grouped together for the purposes of routing. An SNP pool has a strong relationship to Link Ends (See ITU-T Rec. G.852.2).

3.21 Subnetwork Point Pool link (SNPP link): An association between SNPPs on different subnetworks.

3.22 Termination Connection Point (TCP): For the purposes of this Recommendation, a Termination Connection Point represents the output of a Trail Termination function or the input to a trail termination sink function. (Note that in ITU-T Rec. G.805 the TCP refers to the binding between two points).

3.23 trail: See ITU-T Rec. G.805.

3.24 Trail Termination Point (TTP): See ITU-T Rec. M.3100. A Trail Termination Point represents the signal state at a TCP.

3.25 transport plane: The Transport Plane provides bidirectional or unidirectional transfer of user information, from one location to another. It can also provide transfer of some control and network management information. The Transport Plane is layered; it is equivalent to the Transport Network defined in ITU-T Rec. G.805.

4 Abbreviations

This Recommendation uses the following abbreviations:

AG	Access Group
CC	Connection Controller
CP	Connection Point
CPS	Connection Point Status
CTP	Connection Termination Point
DCN	Data Communications Network
E-NNI	Logical External Network-Network Interface (reference point)
HOVC	Higher Order Virtual Container
id	identifier
I-NNI	Logical Internal Network-Network Interface (reference point)
LOVC	Lower Order Virtual Container
LRM	Link Resource Manager
PC	Protocol Controller
RC	Routing Controller

SNC Subnetwork Connection

SNP Subnetwork Point

SNPP Subnetwork Point Pool

TCP Termination Connection Point

TTP Trail Termination Point

UML Unified Modelling Language

UNI Logical User-Network Interface (reference point)

VPN Virtual Private Network

5 Overview

The purpose of the Automatic Switched Optical Network control plane is to:

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

– Reconfigure or modify connections that support calls that have previously been set up.

– Perform a restoration function.

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

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

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

Figure 1 provides a high level view of the interactions of the control, management and transport planes for the support of switched connections of a layer network. Also included on this figure is the DCN, which provides the communication paths to carry signalling and management information. The details of the DCN, management plane and the transport plane are outside the scope of this Recommendation. Functions pertaining to the control plane are described in this Recommendation.

The control plane supports connection set-up/teardown as a result of a user request (switched connection) and a management request (soft permanent connection). In addition a control plane may have to support re-establishing a failed connection (e.g. restoration). Connection state information (e.g. fault and signal quality) is detected by the transport plane and provided to the control plane.

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

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

The control plane will be subdivided into domains that match the administrative domains of the network. The transport plane is also partitioned to match the administrative domains. Within an administrative domain the control plane may be further subdivided, e.g. by actions from the management plane. This allows the separation of resources into, for example, domains for geographic regions, that can be further divided into domains that contain different types of equipment. Within each domain, the control plane may be further subdivided into routing areas for scalability, which may also be further subdivided into sets of control components. The transport plane resources used by ASON will be partitioned to match the subdivisions created within the control plane.

The interconnection between domains, routing areas and, where required, sets of control components is described in terms of reference points. The exchange of information across these reference points is described by the multiple abstract interfaces between control components. The physical interconnection is provided by one or more of these interfaces. A physical interface is provided by mapping an abstract interface to a protocol. Multiple abstract interfaces may be multiplexed over a single physical interface. The reference point between an administrative domain and an end user is the UNI. The reference point between domains is the E-NNI. The reference point within a domain between routing areas and, where required, between sets of control components within routing areas is the I-NNI. These reference points are further described in clause 8.

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

The interactions between the control plane and the layer networks of the transport plane, and any changes in the interaction between the management plane and the transport plane resulting from the addition of the control plane for the:

- management of connections;
- configuration of trail terminations within a layer network;
- configuration of connection monitors;
- client layer to request or release capacity in the server layer.

will be described separately. Only the management of connections within a layer network is included within this version of this Recommendation.

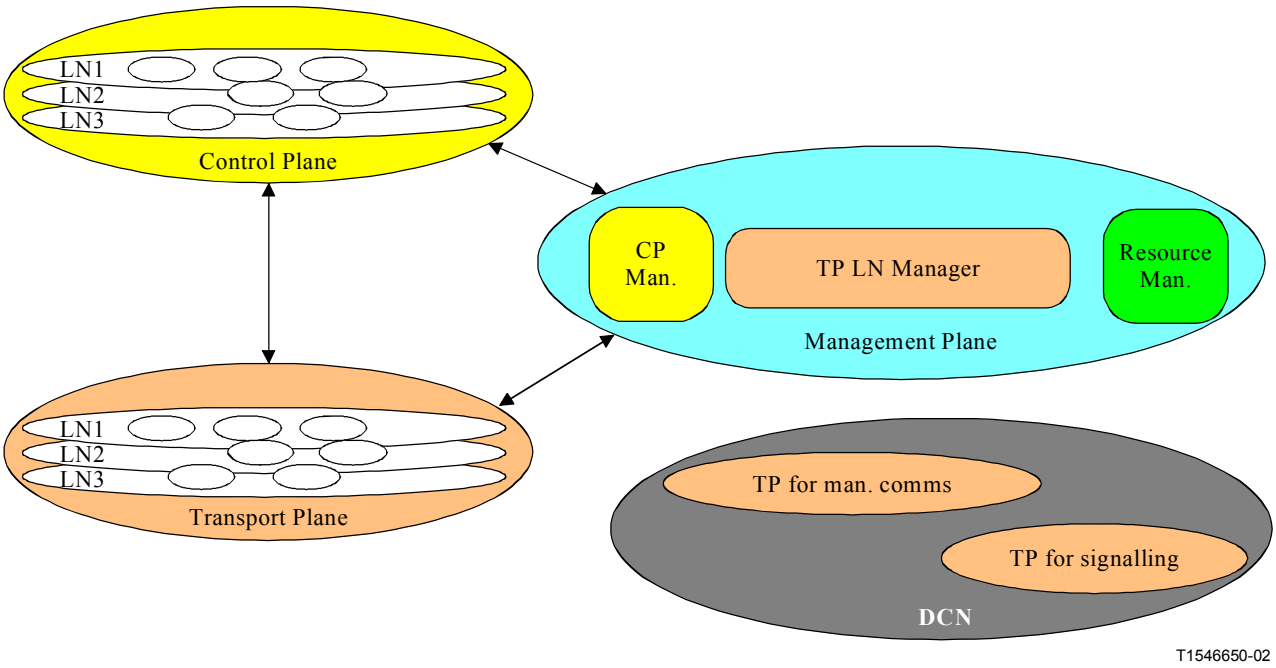


Figure 1/G.8080/Y.1304 – Relationship between architectural components

5.1 Call and connection control

Call and connection control are treated separately in this Recommendation. This has the advantage of reducing redundant call control information at intermediate (relay) connection control nodes, thereby removing the burden of decoding and interpreting the entire message and its parameters. Call control can therefore be provided at the ingress to the network or at gateways and network boundaries. As such the relay bearer needs only provide the procedures to support switching connections.

5.1.1 Call control

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

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

– Separation of the call information into parameters carried by a single call/connection protocol.

– Separation of the state machines for call control and connection control, whilst signalling information in a single call/connection protocol.

– Separation of information and state machines by providing separate signalling protocols for call control and connection control.

Call control must provide coordination of connections (in a multi-connection call) and the coordination of parties (multiparty 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 established. A call may exist without any connections (facilitating complex connection rearrangements).

A call can be considered to have three phases:

Establishment

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

Active

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

Release

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

5.1.2 Call admission control

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

The Originating Call admission function is responsible for checking that a valid called user name and parameters has been provided. The service parameters are checked against a Service Level Specification (a set of parameters and values agreed between Network Operator and Customer for a particular service indicating the '*scope*' of the service). If necessary, these parameters may need to be renegotiated with the originating user. The scope of this negotiation is determined by policies derived from the original Service Level Specification, which itself is derived from the Service Level Agreement (the service contract between a Network Operator and a Customer that defines global responsibilities between them).

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

5.1.3 Connection control

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

5.1.4 Connection admission control

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

5.1.5 Relationship between call state and connection state

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

6 Transport resources and their organization

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

6.1 Transport entities

For the purpose of managing connections within a layer network, the underlying transport plane resources are represented by a number of entities in the control plane. Figure 2 illustrates the relationship between the transport resources described in ITU-T Rec. G.805, the entities that represent these resources for the purposes of network management (as described in ITU-T Rec. M.3100) and the view of the transport resources as seen by the control plane.

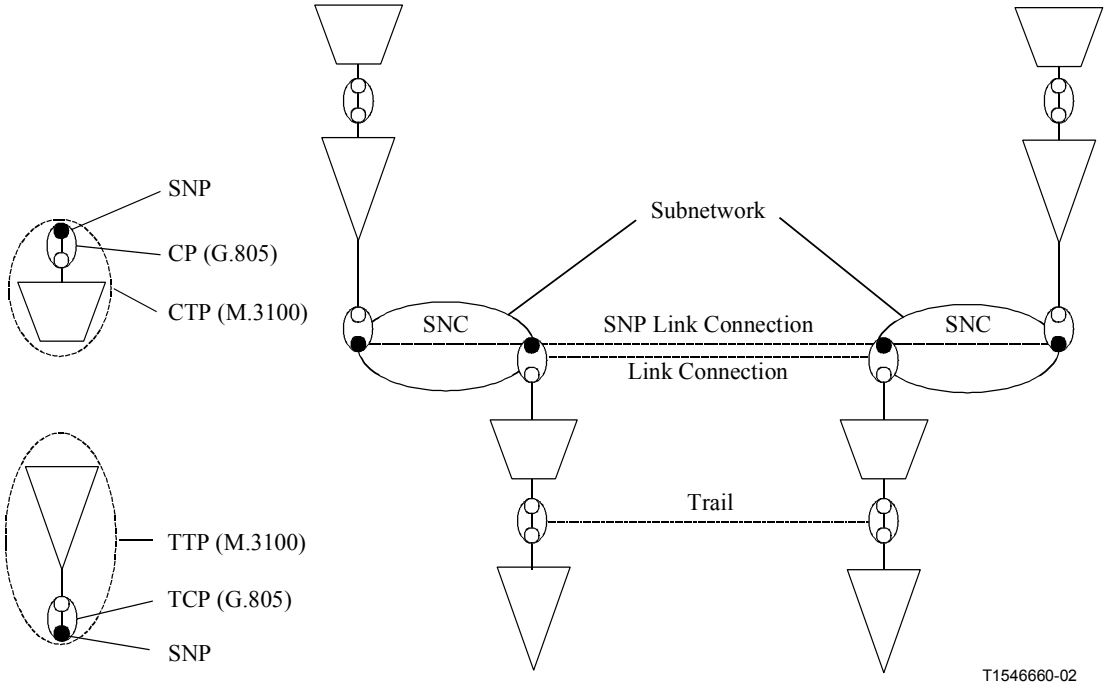


Figure 2/G.8080/Y.1304 – Relationship between architectural entities in the transport plane, management plane and the control plane

An SNP has a number of relationships with other SNPs:

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

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

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

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

The SNP has the following potential states that may be of interest to the control plane for the purposes of connection management:

– Available: Adaptation activated, CTP exists and link connection exists.

– Potential: Adaptation not activated, CTP does not exist.

– Provisioned: In use by this partition of the subnetwork.

– Busy: The underlying transport resource is being used by another layer network or SNP in another subnetwork.

Variable adaptation functions

A number of transport systems support variable adaptation, whereby a single server layer trail may dynamically support different multiplexing structures.

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

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

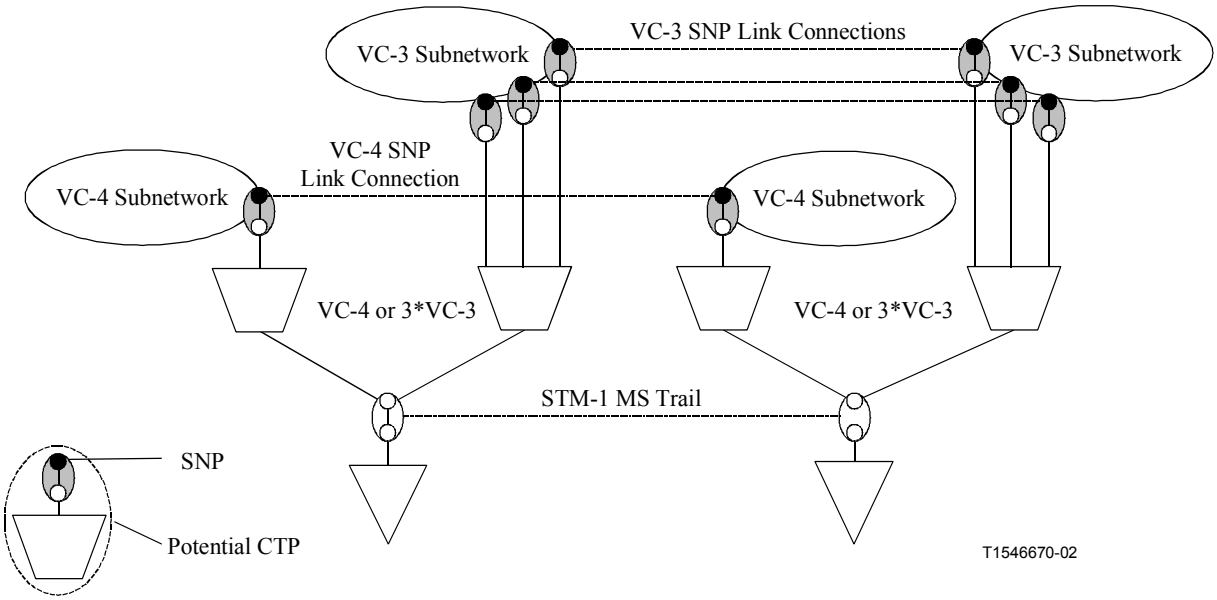


Figure 3/G.8080/Y.1304 – Example of variable adaptation (STM-1 trail supporting both 3 × VC-3 or 1 × VC-4)

Link resources shared between VPNs

In this Recommendation, a VPN is defined as a set of virtually dedicated transport resources, supporting a closed user group, over transport links that are shared between multiple users.

Connectivity on a link that is shared between VPNs can be modelled by creating an SNP for each of the shared CPs in each VPN subnetwork. When a particular SNP is allocated in one VPN subnetwork, the SNPs representing the same resources in other VPN subnetworks become Busy. Figure 4 shows an example of two VPNs, each with one available and one potential CP on a shared link that can support a total of three CPs.

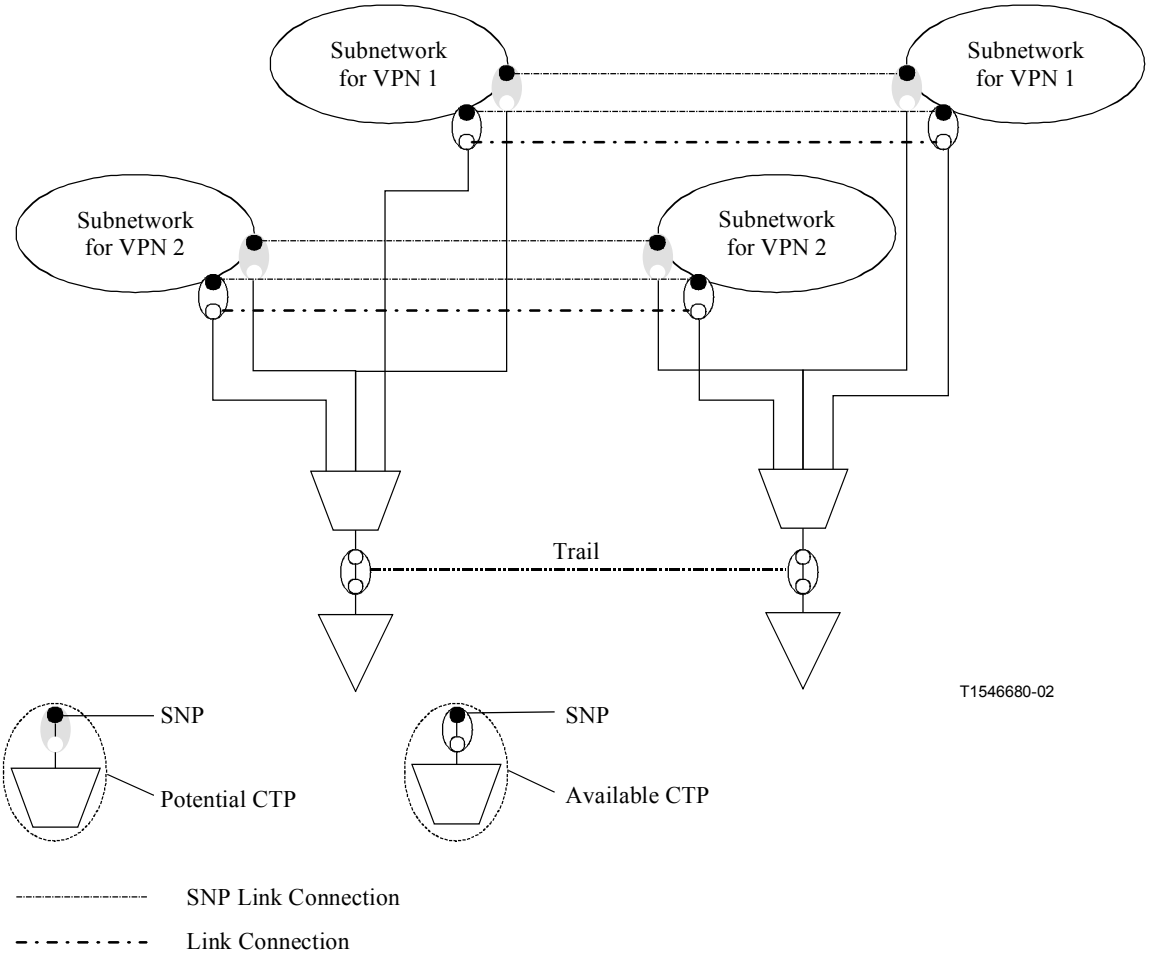


Figure 4/G.8080/Y.1304 – Allocation of link resources between VPNs

6.2 Routing areas

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

Where an SNPP link crosses the boundary of a routing area, all the routing areas sharing that common boundary use a common SNPP id to reference the end of that SNPP link. This is illustrated in Figure 5.

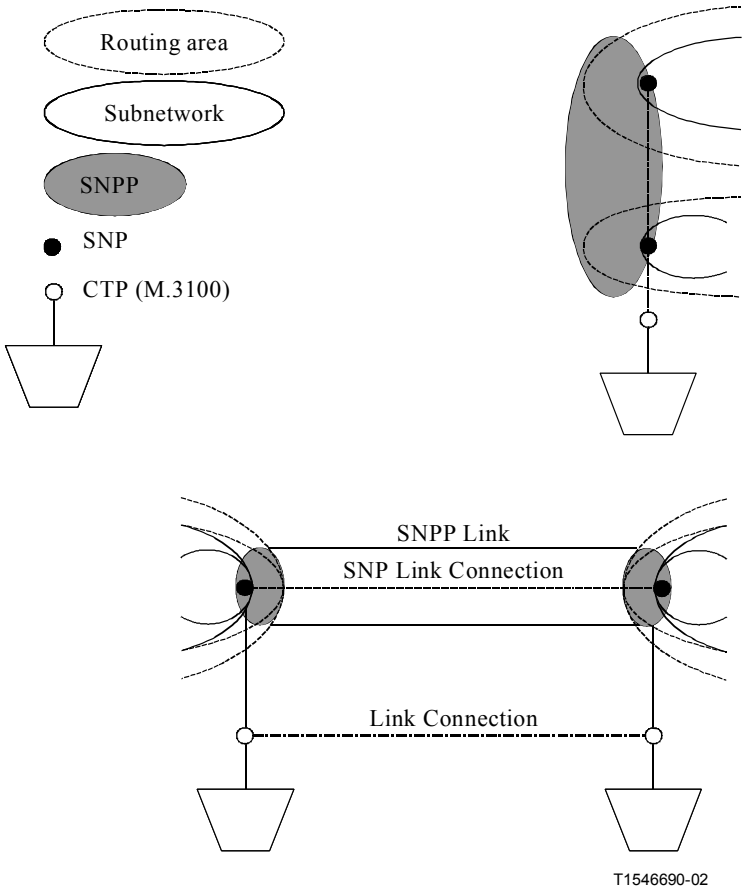


Figure 5/G.8080/Y.1304 – Relationship between routing areas, subnetworks, SNPs and SNPP

6.3 Topology and discovery

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

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

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

The link information (e.g. the constituent link connections and the names of the CTP pairs) is then used to configure the LRM instances (as described in 7.3.3) associated with the SNPP link. Additional characteristics of the link, based on parameters of the link connections, may also be provided. The LRMs at each end of the link must establish a control plane adjacency that corresponds to the SNPP link. The interface SNPP ids may be negotiated during adjacency discovery or may be provided as part of the LRM configuration. The Link Connections and CTP names are then mapped to interface SNP ids (and SNP Link Connection names). In the case where both ends of the link are within the same routing area the local and interface SNPP id and the local and interface SNP ids may be identical. Otherwise, at each end of the link the interface SNPP id is

mapped to a local SNPP id and the interface SNP ids are mapped to local SNP ids. This is shown in Figure 6.

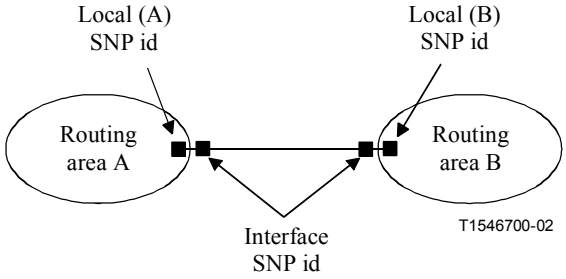


Figure 6/G.8080/Y.1304 – Relationship between local and interface ids

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

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

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

7 Control plane architecture

This clause describes a reference architecture for the control plane that supports the requirements in ITU-T Rec. G.807/Y.1302, identifying its key functional components and how they interact. This flexible reference architecture is intended to enable operators to support their internal business and managerial practices, as well as to bill for services they provide to their customers. The control plane architecture should have the following characteristics:

- Support various transport infrastructures, such as the SONET/SDH transport network, as defined in ITU-T Rec. G.803, and the Optical Transport Network (OTN), as defined in ITU-T Rec. G.872.
- Be applicable regardless of the particular choice of control protocol (i.e. employ a protocol neutral approach that is independent of the particular connection control protocols used).
- Be applicable regardless of how the control plane has been subdivided into domains and routing areas, and how the transport resources have been partitioned into subnetworks.
- Be applicable regardless of the implementation of connection control that may range from a fully distributed to a centralized control architecture.

This reference architecture describes the:

- functional components of the control plane, including abstract interfaces and primitives;
- interactions between call controller components;
- interactions among components during connection set-up;
- functional component that transforms the abstract component interfaces into protocols on external interfaces.

7.1 Notation

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

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

Table 1/G.8080/Y.1304 – Generic interface descriptions

Input interface	Basic input parameters	Basic return parameters

Output interface	Basic output parameters	Basic return parameters

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

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

Component: In this Recommendation, components are used to represent abstract entities, rather than instances of implementation code. They are used to construct scenarios to explain the operation of the architecture. This component is represented as a rectangle with tabs. This is illustrated in Figure 7.

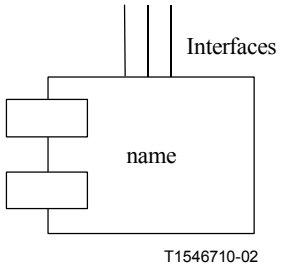


Figure 7/G.8080/Y.1304 – Representation of a component

Generically, every component has a set of special interfaces to allow for monitoring of the component operation, and dynamically setting policies and affecting internal behaviour. These interfaces are not mandatory, and are provided on specific components only where necessary.

Where appropriate, the use of the monitor interface is described in individual component descriptions. Components are not assumed to be statically distributed.

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

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

7.2 Policy and federations

7.2.1 General model of policy

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

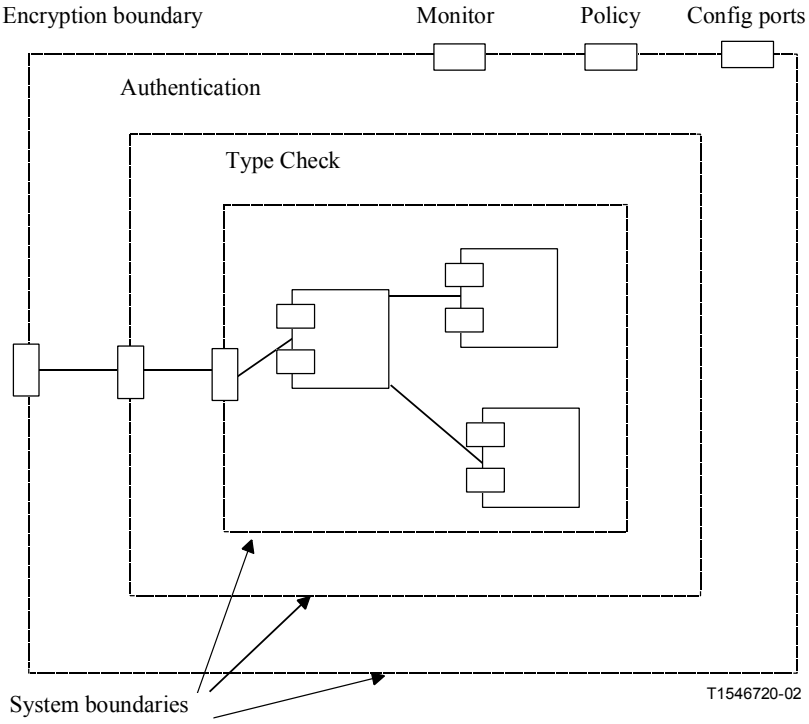


Figure 8/G.8080/Y.1304 – Example of System boundaries related to policy control

In Figure 8, the dashed boxes represent system boundaries, while the closed rectangles on the boundary, called ports, represent Port Controller components.

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

Figure 8 shows an example of how encryption, authentication and type checking may be implemented as a set of three nested Port Controllers, where the policy application order follows the nesting order. The components inside the authentication boundary do not specify encryption or authentication requirements, as these are properties of the component environment. Port Controllers are defined for each independent aspect of port policy, and combined policy is achieved by composition of Port Controllers. This allows the creation of reusable components, which are distinguished by a descriptive prefix. Policy violations are reported via the monitoring port.

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

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

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

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

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

7.2.2 General model of federation

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

There are two types of federation:

– Joint federation model.

– Cooperative model.

In the joint federation case one connection controller, the parent connection controller, has authority over connection controllers that reside in different domains. Where a connection is required that crosses multiple domains the highest-level connection controller (the parent) acts as the coordinator. This connection controller has knowledge of the highest-level connection controllers in each domain. The parent connection controller divides the responsibility for the network connection between the next level connection controllers, with each responsible for its part of the connection. This is illustrated in Figure 9. This model is recursive with a parent connection controller at one level being a child to a parent at a higher level.

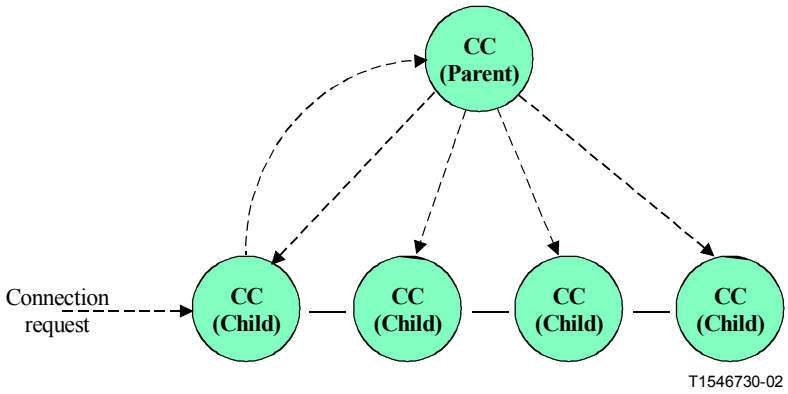


Figure 9/G.8080/Y.1304 – Joint federation model

In the cooperative model, there is no concept of a parent connection controller. Instead when a connection request is made the originating connection controller contacts each of the connection controllers associated with domains of its own volition and there is no overall coordination. The simplest method of achieving this is for the originating connection controller to contact the next connection controller in the chain. This is illustrated in Figure 10, where each connection controller calculates what part of the connection it can provide and what the next connection controller will be. This continues until the connection is provided.

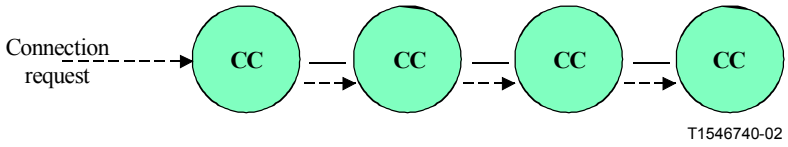


Figure 10/G.8080/Y.1304 – Cooperative federation model

Federation between administrative domains is by means of the cooperative model. In this case all administrative domains are expected to have the capability to federate with other administrative domains. Parent connection controllers within an administrative domain may federate with other parent connection controllers in other administrative domains by means of the cooperative model. An administrative domain may also be subdivided and the choice of federation model employed between domains within an administrative domain can be independent of what happens in another administrative domain. It is therefore possible to combine both federation models to construct large networks as illustrated in Figure 11. The principle described above can also be applied to federations of call controllers.

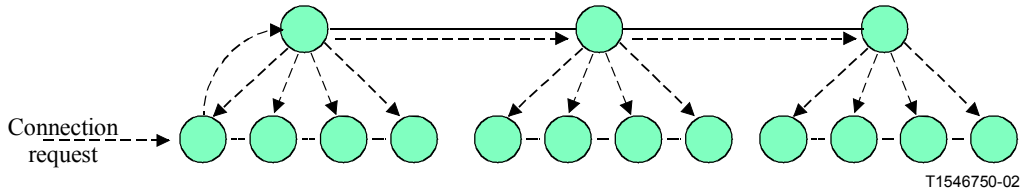


Figure 11/G.8080/Y.1304 – Combined federation model

7.3 Architectural components

The components of the control plane architecture are described in this clause. Components can be combined in different ways, depending upon the required functionality, this is illustrated in 7.5. Each component is described by a brief description of its primary function in this reference architecture. Component interfaces are next provided, and a more detailed description of operation is then given.

7.3.1 Connection controller (CC) component

The connection controller is responsible for coordination among the Link Resource Manager, Routing Controller, and both peer and subordinate Connection Controllers for the purpose of the management and supervision of connection set-ups, releases and the modification of connection parameters for existing connections.

This component services a single subnetwork, and provides the abstract interfaces to other control plane components given in Table 2. The connection controller component is illustrated in Figure 12.

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

Table 2/G.8080/Y.1304 – Connection controller component interfaces (update table based on figures)

Input interface	Basic input parameters	Basic return parameters
Connection Request In	A pair of local SNP names	A subnetwork connection
Peer coordination In	1) A pair of SNP names; or 2) SNP and SNPP; or 3) SNPP pair.	Confirmation signal

Output interface	Basic output parameters	Basic return parameters
Route Table Query	Unresolved route fragment	An ordered set of SNPPs
Link Connection Request	–	A Link Connection (an SNP pair)
Connection Request Out	A pair of local SNP names	A subnetwork connection
Peer coordination Out	1) A pair of SNP names; or 2) SNP and SNPP; or 3) SNPP pair.	Confirmation signal

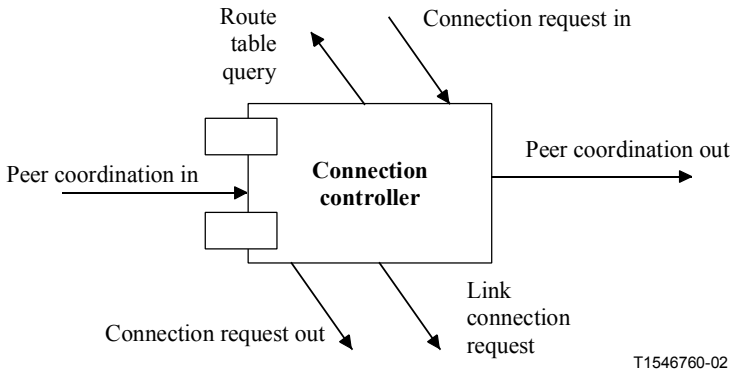


Figure 12/G.8080/Y.1304 – Connection controller component

Connection set-up Operation

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

The first unresolved portion of the route is resolved, via the Route Table Query interface, into a set of links to be traversed, and this new set of links adds to the set. The connection controller inspects the new set of links to see which of these links are available for link connection allocation. Link connections are obtained and their links are removed from the link set. Next, corresponding subnetwork connections are requested from subordinate (i.e. child) Connection Controllers via the Connection Request Out interface. Any unallocated route components are passed on to the next downstream peer Connection Controller. The actual sequence of operations depends on many factors, including the amount of routing information available and the access to particular Link Resource Managers; however, the operation of the Connection Controller is invariant. Connection teardown is an analogous operation to connection setup, except the operations are reversed.

7.3.2 Routing Controller (RC) component

The role of the routing controller is to:

– respond to requests from connection controllers for path (route) information needed to set up connections. This information can vary from end-to-end (e.g. source routing) to next hop;

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

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

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

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

The routing controller has the interfaces provided in Table 3 and illustrated in Figure 13.

Table 3/G.8080/Y.1304 – Routing controller interfaces

Input interface	Basic input parameters	Basic return parameters
Route Table Query	Unresolved route element	Ordered list of SNPPs
Local Topology In	Local topology update	–
Network Topology In	Network topology update	–

Output interface	Basic output parameters	Basic return parameters
Local Topology Out	Local topology update	–
Network Topology Out	Network topology update	–

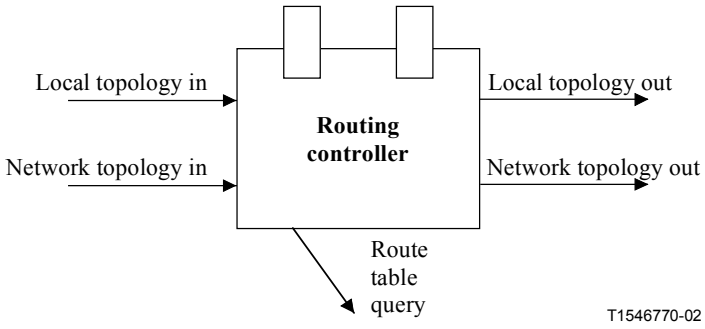


Figure 13/G.8080/Y.1304 – Routing controller component

Route Query interface: This interface accepts an unresolved route element and returns a set of links that are within the domain of responsibility of the routing controller. Forms of replies include, but are not limited to, step-by-step forwarding (next link) and source routing (full path). Examples of look-up results are:

1) Return an egress SNPP on this subnet that is on a path to a given destination SNPP.
2) Return a sequence of subnetworks that form a path between a given source/destination SNPP pair.
3) Return a sequence of subnetworks that form a path between two sets of SNPPs.
4) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair.
5) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair and includes one or more specific SNPPs.
6) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair that is diverse from a given path.

Local Topology interface: This interface is used to configure the routing tables with local topology information and local topology update information. This is the topology information that is within the domain of responsibility of the routing controller.

Network Topology interface: This interface is used to configure the routing tables with network topology information and network topology update information. This is the reduced topology information (e.g. summarized topology) that is outside the domain of responsibility of the routing controller.

7.3.3 Link resource manager (LRMA and LRMZ) component

The LRM components are responsible for the management of an SNPP link; including the allocation and deallocation of SNP link connections, providing topology and status information.

Two LRM components are used – the LRMA and LRMZ. An SNPP link is managed by a pair of LRMA and LRMZ components one managing each end of the link. Requests to allocate SNP link connections are only directed to the LRMA.

The two cases for SNPP link are illustrated in Figure 14.

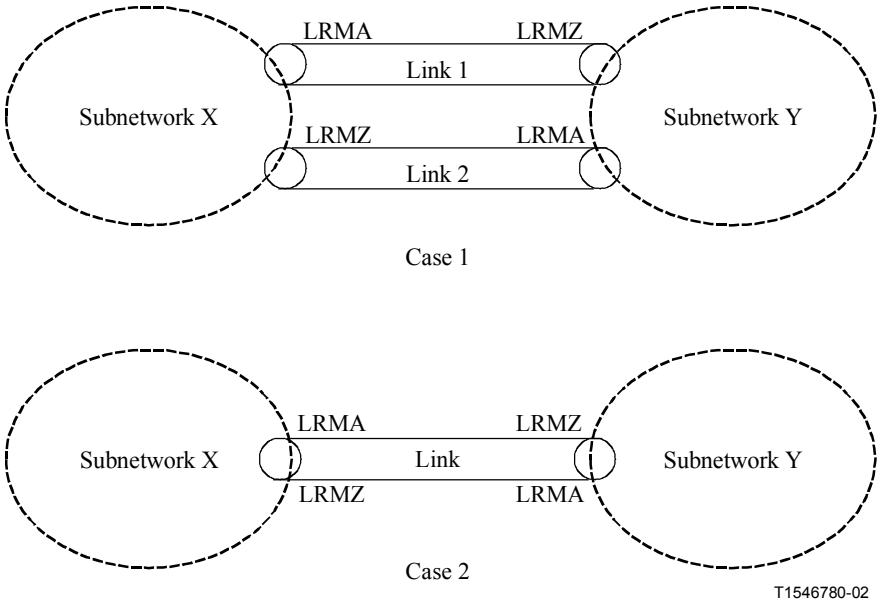


Figure 14/G.8080/Y.1304 – SNPP link cases

In case 1, link 1 is dedicated to connection set-up originating from subnetwork X. Requests for SNP link connections from subnetwork X are directed to the adjacent LRMA for link 1. This LRMA can allocate the SNP link connection without negotiation with the LRMZ for link 1. Similarly link 2 is dedicated to connection set-up requests originating from subnetwork Y. Requests for SNP link connections from subnetwork Y are directed to the adjacent LRMA for link 2. This LRMA can allocate the SNP link connection without negotiation with the LRMZ for link 2.

In case 2 the link is shared between subnetworks X and Y for connection set-up. Requests for SNP link connections from subnetwork X are directed to the adjacent LRMA, since an LRMA component at the far end of the link can also allocate SNP link connections, the LRMA may need to negotiate an allocation with the LRMZ at the far end. A similar process is required for request from subnetwork Y to its adjacent LRMA.

7.3.3.1 LRMA

The LRMA is responsible for the management of the A end of the SNPP link, this includes the allocation and deallocation of link connections, providing topology and status information.

The LRMA component interfaces are provided in Table 4 and illustrated in Figure 15.

Table 4/G.8080/Y.1304 – LRMA component interfaces

Input interface	Basic input parameters	Basic return parameters
SNP link connection request	Request id SNP Id (optional)	Request id SNP id pair or denied
SNP link connection deallocation	SNP id	Confirm or denied
Configuration	Link information	–
Translation	Local id	Interface id

Output interface	Basic output parameters	Basic return parameters
SNP negotiation (Case 2 only)	Request id List of SNP ids	Request id SNP id
SNP release (Case 2 only)	SNP id	Confirm
Topology	Link information	–

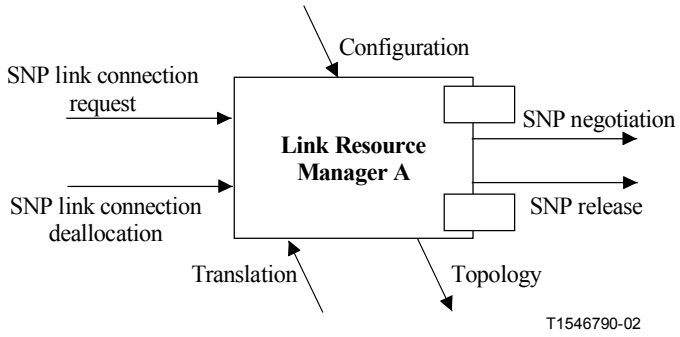


Figure 15/G.8080/Y.1304 – Link Resource Manager A component

Functions

Allocation of link connection

When a request to allocate a link connection is received connection admission is invoked to decide if there is sufficient free resource to allow a new connection. Connection admission can also be decided based on prioritization or on other policy decisions. Connection admission policies are outside the scope of standardization (see ITU-T Rec. G.807/Y.1302).

If there are insufficient resources the request is rejected.

If sufficient resources are available the connection request is allowed to process as described in the two cases below.

– Case 1: Since the SNP link connections are only allocated from one end of the SNPP link the LRMA can select the SNP link connection without interaction with the LRMZ at the far end of the link.

– Case 2: Since the SNP link connections may be used by the LRMA at either end of the SNPP link the LRMA passes a list of usable SNP ids to the LRMZ. The LRMZ (in cooperation with its local LRMA) selects one of the SNPs and returns the id to the originating LRMA.

Deallocation of a link connection

When a request to deallocate a SNP link connection is received the corresponding SNP is marked as available. In case 2 the associated LRMZ is informed.

Interface to local id translation

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

Topology

This function provides the link topology using the interface SNPP ids and the contained SNP ids.

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

7.3.3.2 LRMZ

The LRMZ is responsible for the management of the Z end of the SNPP link, this includes providing topology information.

The LRMZ component interfaces are provided in Table 5 and illustrated in Figure 16.

Table 5/G.8080/Y.1304 – LRMZ component interfaces

Input interface	Basic input parameters	Basic return parameters
SNP negotiation In (Case 2 only)	Request id List of SNP ids	Request id SNP id or denied
SNP deallocation (case 2 only)	SNP id	Confirmation
Configuration	Link information	–
Translation	Local id	Interface id

Output interface	Basic output parameters	Basic return parameters
Topology	Link information	–

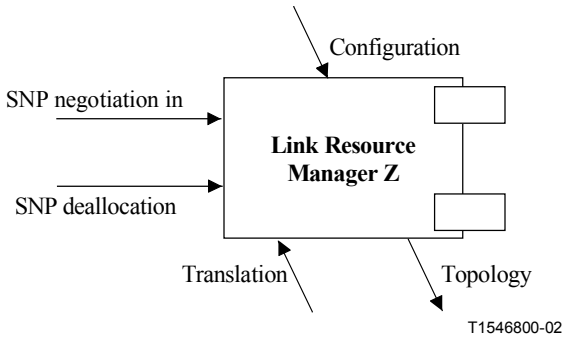


Figure 16/G.8080/Y.1304 – Link Resource Manager Z component

Functions

Allocation of SNP (only used for case 2)

When a list of usable SNP ids is received one is selected and returned.

Deallocation of SNP (only used for case 2)

When the associated LRMA indicates that a SNP has been deallocated the SNP is marked as available.

Interface to local id translation

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

Topology

This function provides the link topology using the interface SNPP ids.

7.3.4 Traffic Policing (TP) component

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

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

7.3.5 Call controller components

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

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

– A network call controller: A network call controller provides two roles, one for support of the calling party and the other to support the called party.

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

7.3.5.1 Calling/called party call controller

The role of this component is:

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

This component has the interfaces provided in Table 6. The calling/called party call controller component is illustrated in Figure 17.

Table 6/G.8080/Y.1304 – Calling/called party call controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Call Accept	Call Source and Destination Identifiers	Confirmation or Rejection of call request
Call Teardown In	Call Source and Destination Identifiers	Confirmation of call teardown

Output interface	Basic output parameters	Basic return parameters
Call Request	Call Source and Destination Identifiers	Confirmation or Rejection of call request
Call Teardown Out	Call Source and Destination Identifiers	Confirmation of call teardown

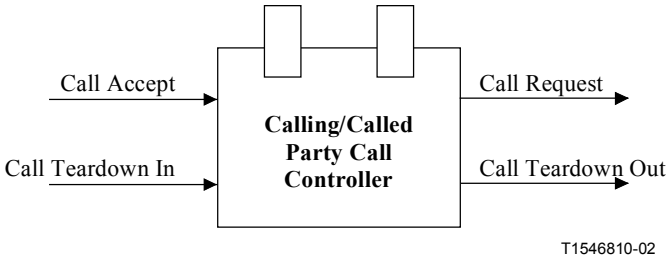


Figure 17/G.8080/Y.1304 – Calling/called party call controller component

Call Request: This interface is used to place requests for set-up, maintenance and cessation 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 Teardown: This interface is used to place, receive and confirm teardown requests.

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

7.3.5.2 Network call controller

The role of this component is:

- processing of incoming call requests;
- generation of outgoing call requests;

– generation of call termination requests;
– processing of call termination requests.
– call admission control based on validation of call parameters, user rights and access to network resource policy;
– call state management.

This component has the interfaces provided in Table 7, below, and illustrated in Figure 18.

Table 7/G.8080/Y.1304 – Network call controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Call Request Accept	Call Source and Destination Identifiers	Confirmation or rejection of call request
Network Call Coordination In	Call Source and Destination Identifiers	Confirmation or rejection
Call Teardown In	Call Source and Destination Identifiers	Confirmation of call teardown

Output interface	Basic output parameters	Basic return parameters
Call Indication	Call Source and Destination Identifiers	Confirmation of rejection of call request
Connection Request Out	Call Source and Destination Identifiers	A pair of SNPs
Network Call Coordination Out	Call Source and Destination Identifiers	Confirmation or rejection of call request
Directory Request	Local name	Call Source/Destination Identifier
Policy Out	Call parameters	Accept or Rejection of Call
Call Teardown Out	Call Source and Destination Identifiers	Confirmation of call teardown

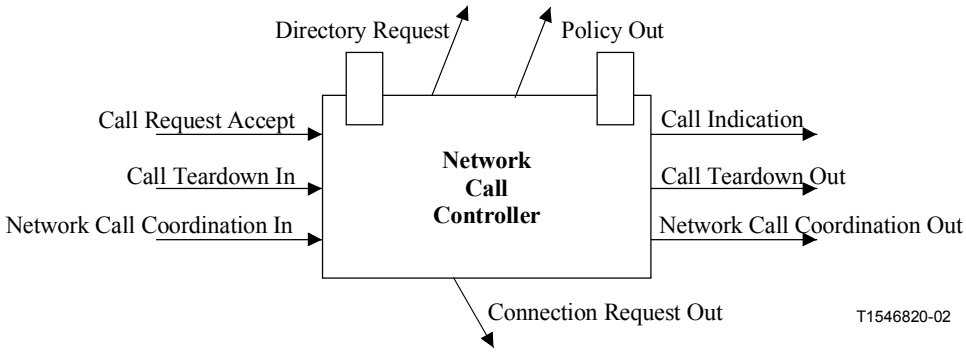


Figure 18/G.8080/Y.1304 – Network call controller component

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 request.

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

Directory Request: This interface is used to get a local name from a source/destination identifier.

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

Call Teardown In/Out: These interfaces are used to place, receive and confirm teardown requests.

Policy Out: This interface provides policy checking.

The role of call admission control in the Calling Party Network Call Controller is to check that a valid called user name and service parameters have been provided. The service parameters are checked against a Service Level Specification. If necessary, these parameters may need to be renegotiated with the Calling Party call controller. The scope of this negotiation is determined by policies derived from the original Service Level Specification, which itself is derived from the Service Level Agreement.

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

7.3.5.3 Call controller interactions

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

Switched connections: The calling party call controller (associated with an end terminal) interacts with the network call controller to form an incoming call and the network call controller interacts with the called party call controller (associated with an end terminal) to form an outgoing call. The network call controller interacts with the connection controllers to provide the call. An example of this interaction is illustrated in Figure 19. It should be noted that the calling/called party call controllers have no direct interaction with the connection controller.

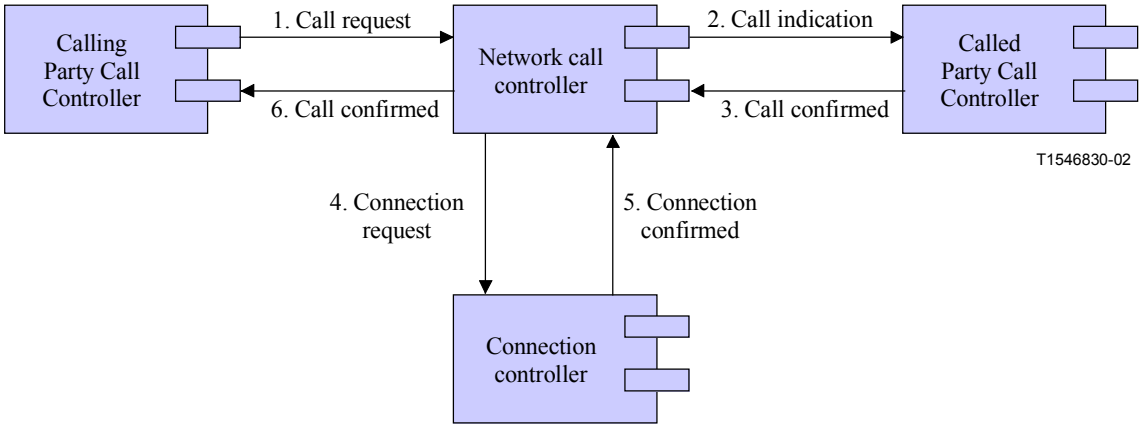


Figure 19/G.8080/Y.1304 – Called/calling party call controller interaction for switched connections: example 1

Figure 19 shows the situation whereby the called party call controller accepts the call, prior to the ingress network call controller requesting the connection. It is also valid to define the interaction such that the connection set-up follows the call, as is illustrated in Figure 20.

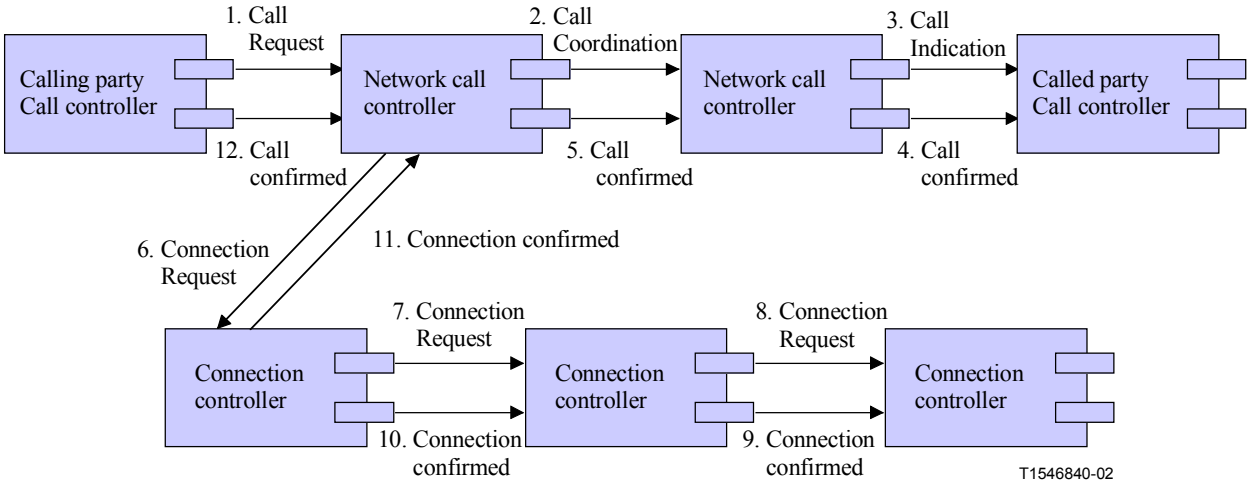


Figure 20/G.8080/Y.1304 – Called/calling party call controller interaction for switched connections: example 2

Soft permanent connections: The network management system is considered to contain both the calling/called party and network call controllers. The management system issues a command to the calling party call controller that initiates the call and receives confirmation of the call set-up. This represents a null call with no service. There are no call/connection protocols between the network management system and the call control. This interaction is illustrated in Figure 21.

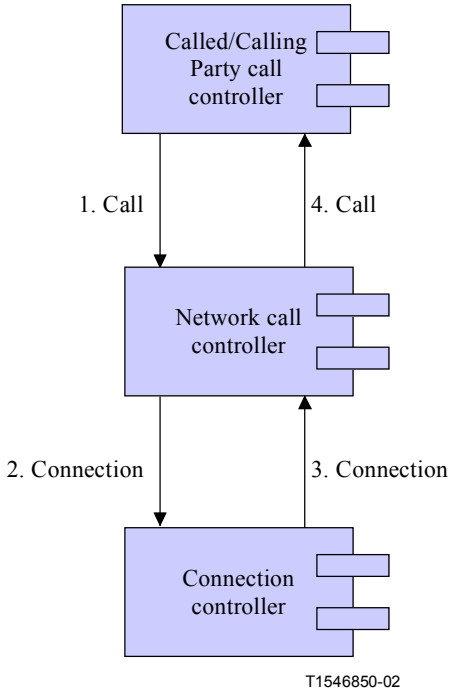
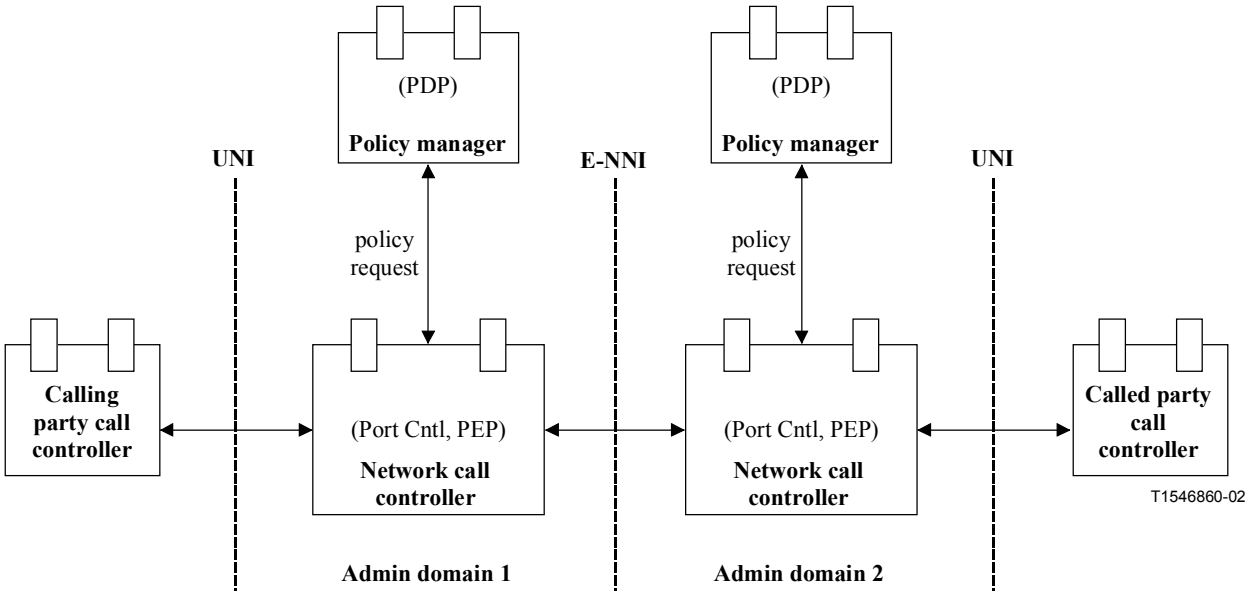


Figure 21/G.8080/Y.1304 – Call controller interactions for soft permanent connections

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

Figure 22 indicates an example of the interactions necessary to support call admission control policy between network call controllers.



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

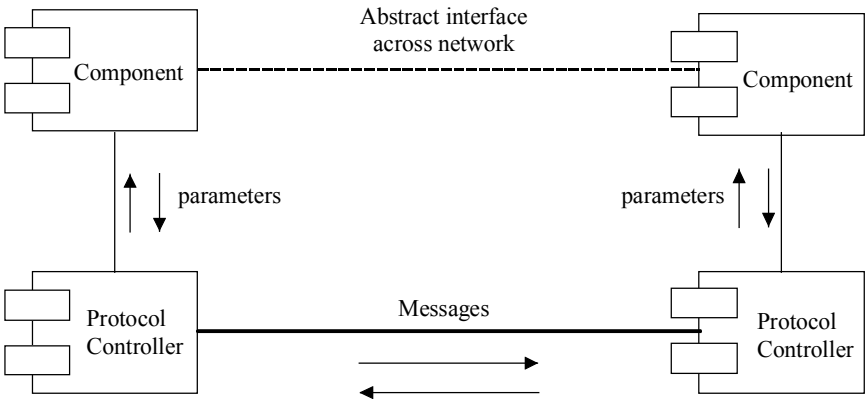
Figure 22/G.8080/Y.1304 – Example of call admission control policy interactions

7.4 Protocol controller (PC) components

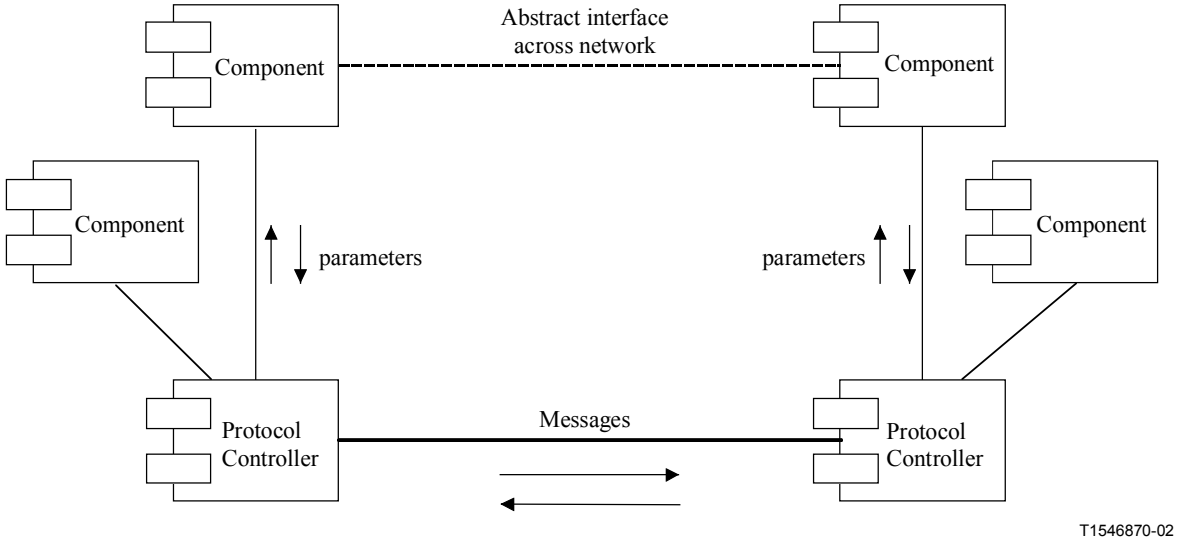
The Protocol Controller provides the function of mapping the parameters of the abstract interfaces of the control components into messages that are carried by a protocol to support interconnection via an interface. Protocol Controllers are a subclass of Policy Ports, and provide all the functions associated with those components. In particular, they report protocol violations to their monitoring ports. They may also perform the role of multiplexing several abstract interfaces into a single protocol instance as shown in Figure 23. The details of an individual protocol controller are in the realm of protocol design, though some examples are given in this Recommendation.

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

Signalling primitives are passed between the connection controller and the protocol controller, which is semantically transparent to the messaging primitives as this results in external protocol messages and vice versa. Signalling messages are passed between the two protocol controllers. This is illustrated in Figure 24.

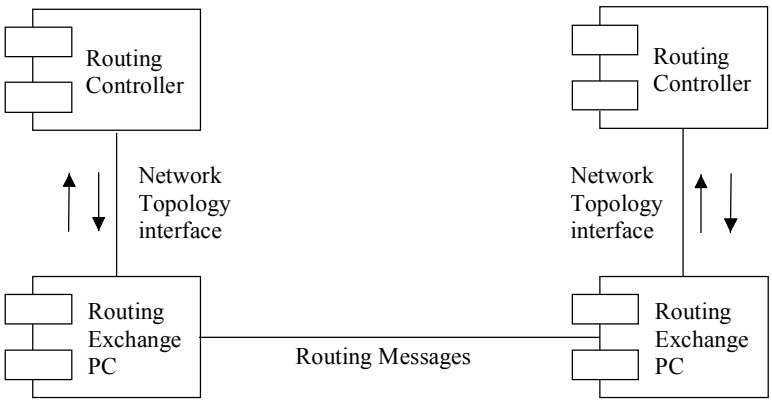


a) Generic use of a Protocol Controller

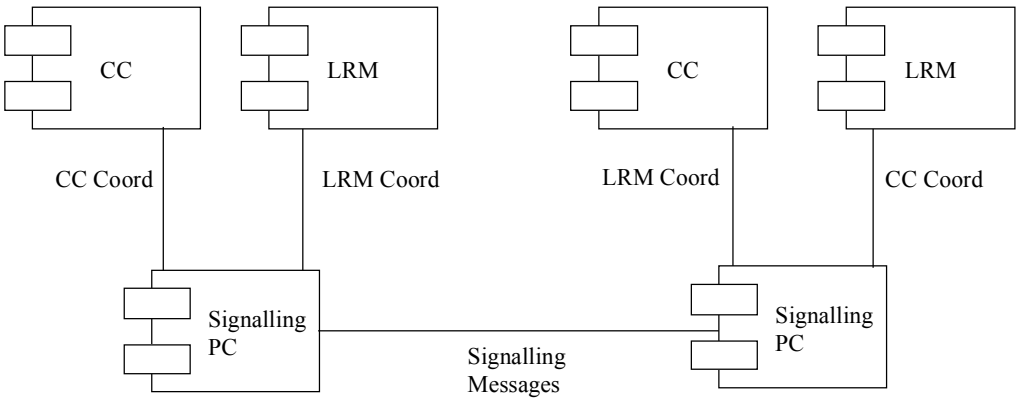


b) Generic multiplexing of different primitive streams into a single protocol

Figure 23/G.8080/Y.1304 – Protocol controller



a) Routing table exchange using routing exchange PC



b) Multiplexing of LRM and CC coordination using signalling PC

T1546880-02

Figure 24/G.8080/Y.1304 – Examples of protocol controller use

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

– Route table update messages via a routing exchange protocol controller (shown in Figure 24 a));

– Link resource manager coordination messages (where appropriate as in available bit rate connections) via a link resource manager protocol controller;

– Connection control coordination messages via a connection controller protocol controller, shown in Figure 24 b). Note that the LRM and CC coordination interfaces may be multiplexed over the same protocol controller.

7.5 Component interactions for connection set-up

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

Three basic forms of algorithm for dynamic path control can be distinguished, hierarchical, source routing and step-by-step routing as shown in the following figures. The different forms of path control result in a different distribution of components between nodes and relationships between these connection controllers.

7.5.1 Hierarchical Routing

In the case of Hierarchical Routing, as illustrated in Figure 25, a node contains a routing controller, connection controllers and link resource managers for a single level in a subnetwork hierarchy. This uses the decomposition of a layer network into a hierarchy of subnetworks (in line with the concepts described in ITU-T Rec. G.805). Connection controllers are related to one another in a hierarchical manner. Each subnetwork has its own dynamic connection control that has knowledge of the topology of its subnetwork but has no knowledge of the topology of subnetworks above or below itself in the hierarchy (or other subnetworks at the same level in the hierarchy).

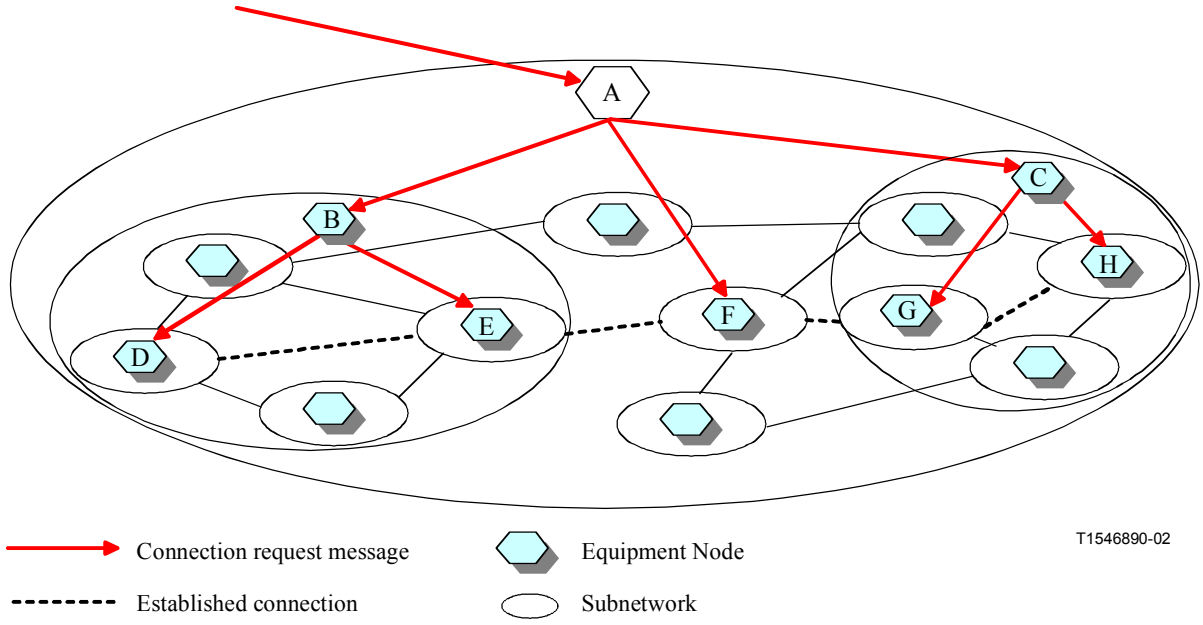


Figure 25/G.8080/Y.1304 – Hierarchical signalling flow

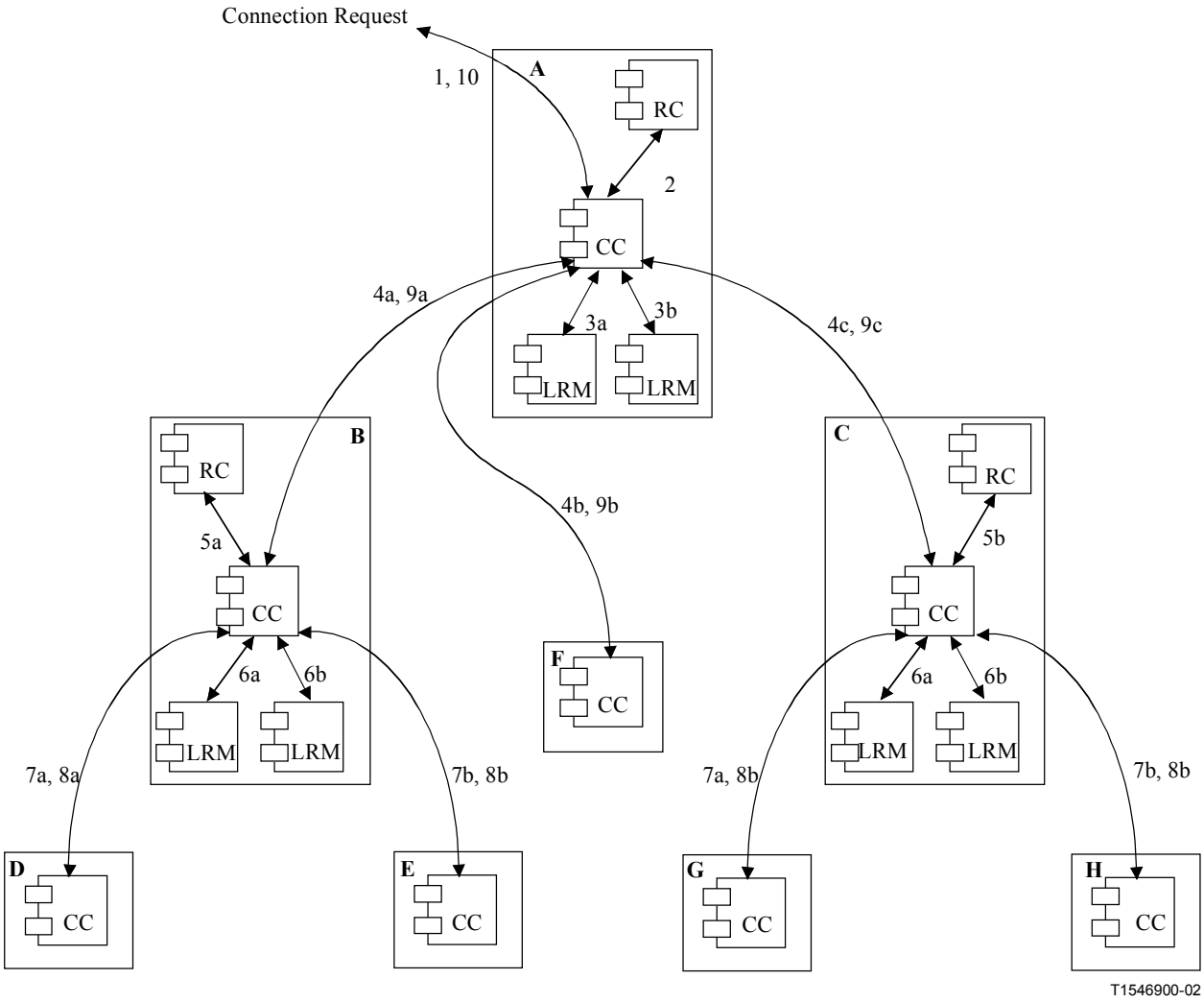


Figure 26/G.8080/Y.1304 – Hierarchical routing interactions

In Figure 26, the detailed sequence of operations involved in setting up a connection using hierarchic routing is described. The steps involved are listed below:

1) A connection request arrives at the Connection Controller (CC), specified as a pair of SNPs at the edge of the subnetwork.

2) The Routing Component (RC) is queried (using the Z end SNP) and returns the set of Links and Subnetworks involved.

3) Link Connections are obtained (in any order, i.e. 3a, or 3b in Figure 26) from the Link Resource Managers (LRM).

4) Having obtained link connections (specified as SNP pairs), subnetwork connections can be requested from the child subnetworks, by passing a pair of SNPs. Again, the order of these operations is not fixed, the only requirement being that link connections are obtained before subnetwork connections can be created. The initial process now repeats recursively.

5) The Child Routing controllers now resolve a route between the SNPs specified.

6) Link Connections are obtained (in any order) from the Link Resource Managers (LRM).

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

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

7.5.2 Source and step-by-step routing

While similar to hierarchical routing, for source routing, the connection control process is now implemented by a federation of distributed connection and routing controllers. The significant difference is that connection controllers operate on Routing Areas whereas they operate on subnetworks in the hierarchical case. The signal flow for source (and step-by-step) routing is illustrated in Figure 27.

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

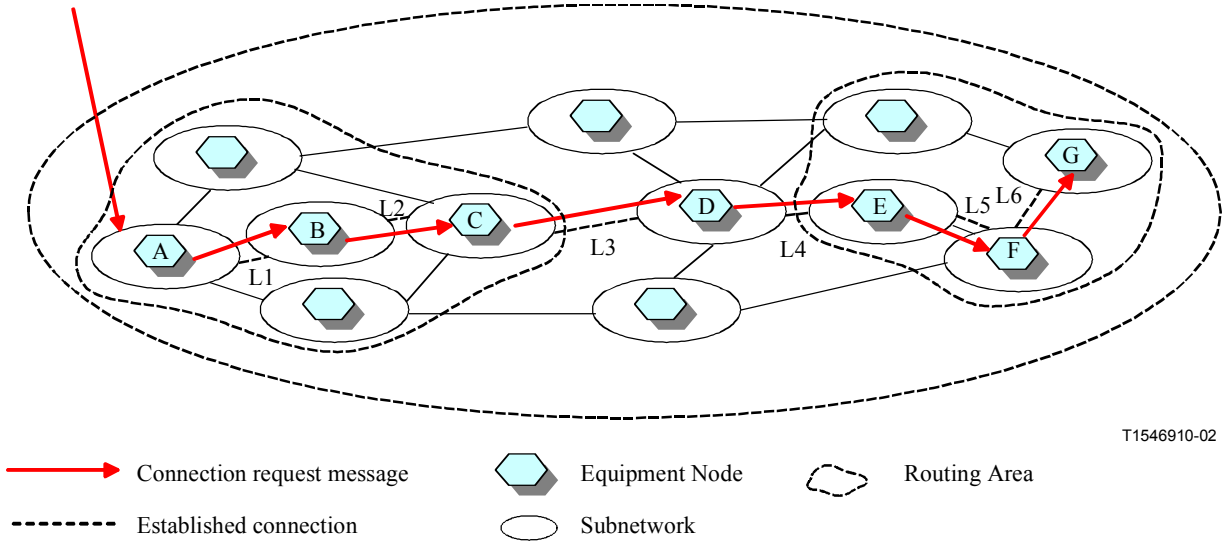


Figure 27/G.8080/Y.1304 – Source and step-by-step signalling flow

Source routing

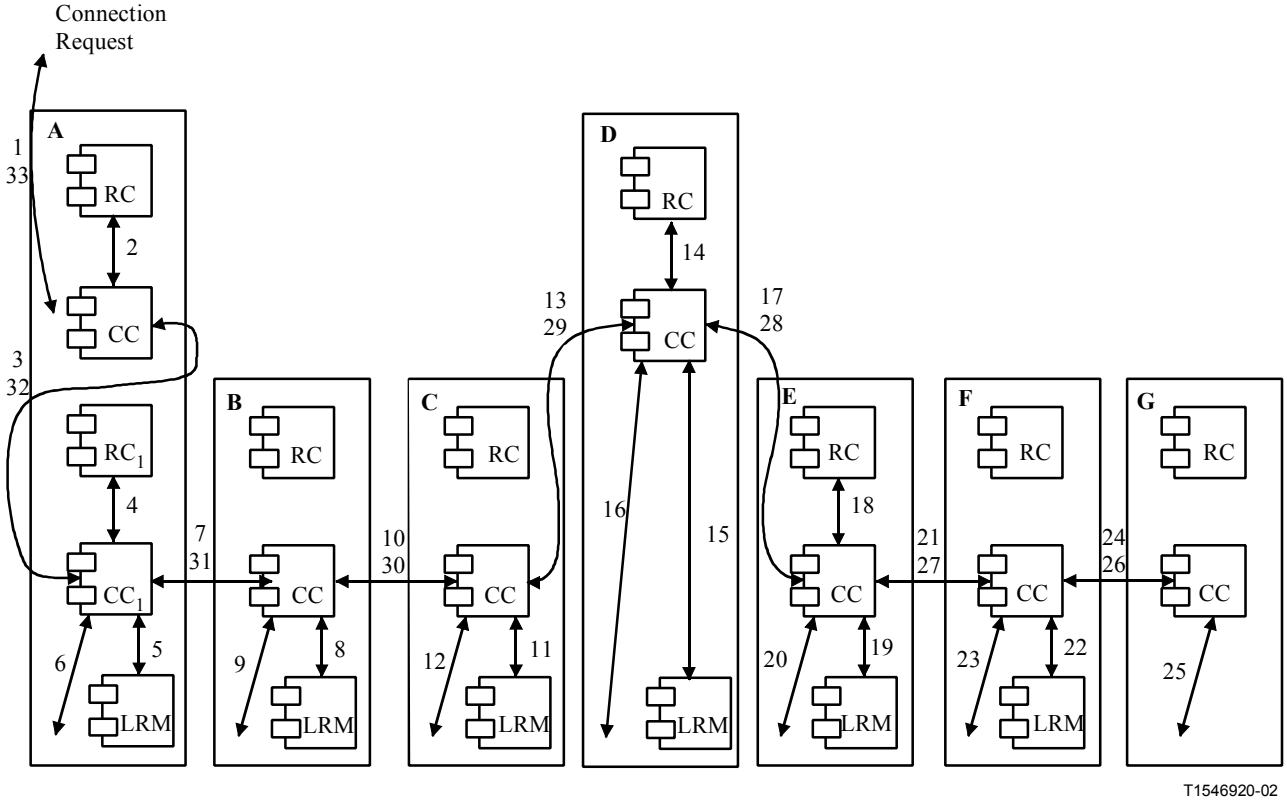


Figure 28/G.8080/Y.1304 – Source routing interactions

In the following steps we describe the sequence of interactions shown in Figure 28. The following notation is used: X_A represents the component at the highest level in Node A, X_{An} represents the component that is at the next nth highest level in Node A.

1) A connection request arrives at the Connection Controller (CC_A), specified as a pair of names (A and Z) at the edge of the subnetwork.

2) The Routing Component (RC_A) is queried (using the Z end SNP) and returns the egress link, L3.

3) As CC_A does not have access to the necessary Link Resource Manager (LRM_C), the request (A, L3, Z) is passed on to a peer CC_{A1}, which controls routing through this Routing Area.

4) CC_{A1} queries RC_{A1} for L3 and obtains a list of additional links, L1 and L2.

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

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

7) The request, now containing the remainder of the route (L2, L3 and Z), is forwarded to the next peer CC_B.

8) LRM_B controls L2, so a link connection is obtained from this link.

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

10) The request, now containing the remainder of the route (L3 and Z), is forwarded to the next peer CC_C.

11) LRM_C controls L3, so a link connection is obtained from this link.

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

13) The request, now containing the remainder of the route (Z), is forwarded to the next peer CC_D.

14) CC_D queries RC_D for Z and obtains link L4.

15) LRM_D controls L4, so a link connection is obtained from this link.

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

17) The request, now containing the remainder of the route (Z), is forwarded to the next peer CC_E.

18) CC_E queries RC_E for Z and obtains links L5 and L6.

The process of connecting across the next Routing Area (i.e. steps 19 to 25 in Figure 28) is identical to that already described. Events 26 to 33 describe the flow of confirmation signals to the connection originator.

Step-by-step routing

In this form of routing there is further reduction of routing information in the nodes, and this places restrictions upon the way in which routing is determined across the subnetwork. Figure 29 applies to the network diagram of Figure 27.

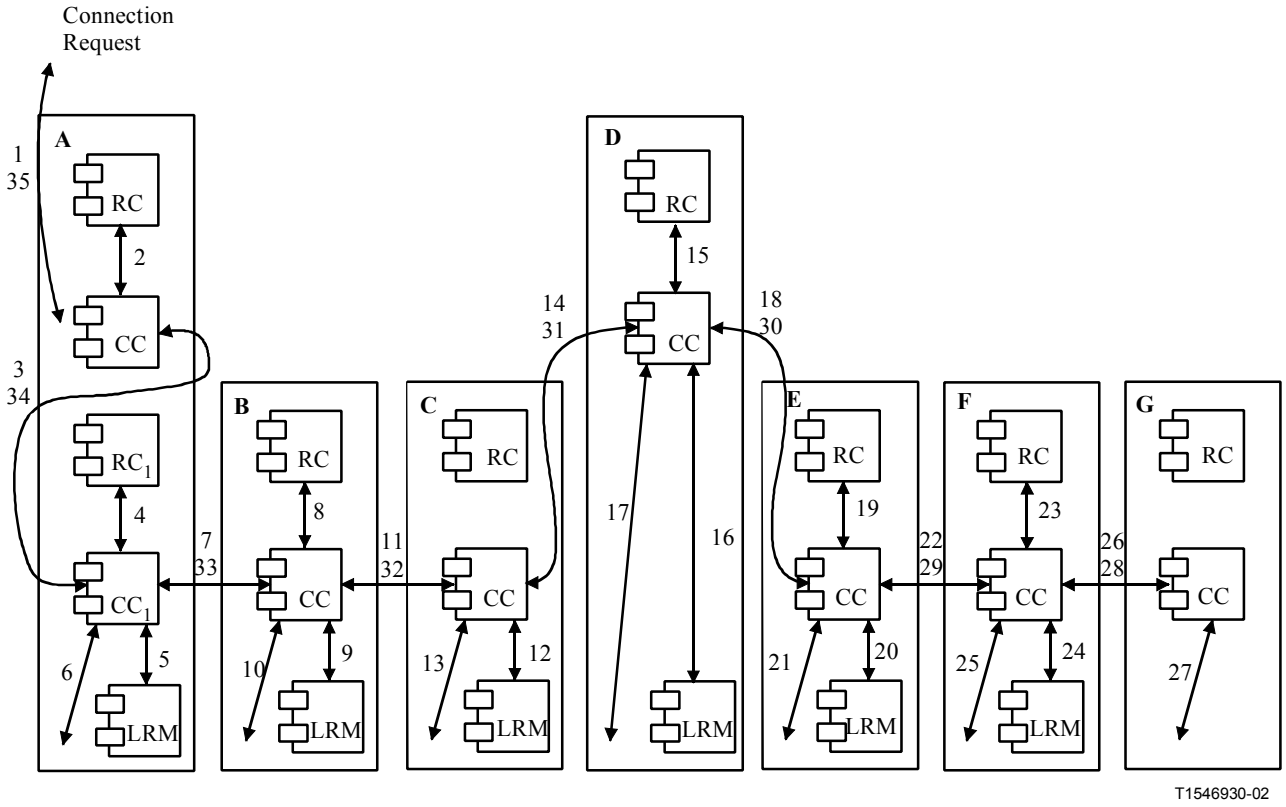


Figure 29/G.8080/Y.1304 – Step-by-step routing

The process of step-by-step routing is identical to that described for Source Routing, with the following variation: Routing Controller RC_{A1} can only supply link L1, and does not supply link L2 as well. CC_B must then query RC_B for L2 in order to obtain L2. A similar process of obtaining one link at a time is followed when connecting across the second Routing Area.

8 Reference points

ITU-T Rec. G.807/Y.1302 defines various logical interfaces (i.e. reference points) within a typical transport network where signalling/routing information is exchanged. Reference points may be supported by multiple interfaces. These reference points are the UNI, the I-NNI and the E-NNI. It is important to recognize that there will be multiple domains within the ASON and that the UNI and E-NNI in particular will be used for inter-domain control signalling. The following clauses describe the specific functionalities that need to be carried across the various reference points (UNI, I-NNI and E-NNI) and how they differ.

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

8.1 UNI

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

- Call Control.
- Resource Discovery.
- Connection Control.
- Connection Selection.

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

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

8.2 I-NNI

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

- Resource Discovery.
- Connection Control.
- Connection Selection.
- Connection Routing.

8.3 E-NNI

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

- Call Control.
- Resource Discovery.
- Connection Control.
- Connection Selection.
- Connection Routing.

9 Network management of control plane entities

There is interaction between the control plane and management plane as described in clause 5. This clause identifies some management capabilities that may impact interactions between the management and control planes. Specifically, such management capabilities may include:

1) Creation and deletion of a connection.

2) Division of network resources between those visible to the control plane and those visible to the management plane.

3) Assignment of capacity to a particular player to create private networks.

4) Assignment of unique identifiers to CTPs and creation of a binding between the CTP and its associated SNPs.

5) Provision of configuration and policy information to the address screening and closed user groups (CUG) functions, if either are present in the control plane.

6) Setting and modification of the signalling system parameters, such as time-outs (e.g. call set-up time-out), thresholds, congestion control mechanisms, maximum number of allowed connections, maximum signalling load (above which the signalling processor denies call establishment requests, etc.).

7) In the event of routing occurring in the management plane (centralized):
   - calculation of the route for permanent connections and use of management protocols for connection management;
   - calculation of the route for soft permanent connections and providing the control plane with an explicit route.

8) Measurement of call performance. The parameters may include:
   - call request rates (arrival rates);
   - circuit utilization;
   - call holding time;
   - holding times (average holding time of connections multiplied by the request rate indicates the offered load in Erlangs);
   - statistical averages computed across the total number of connection requests over a period of time.

9) Management of call admission control.

10) Determination of the maximum number of connections that can be supported by a network element and to set, where appropriate, the maximum number to be supported.

11) Distinguishing changes in the state of connections due to management or control plane actions and those resulting from network failures and suppress or generate alarms as appropriate.

12) Setting or modifying survivability priority levels or Quality of Service (QoS) contract levels for all connections associated with a given "performance class".

13) Assigning the maximum value of a connection identifier on a link where appropriate, setting traffic management controls either manually resulting from a specific input or automatically in response to internal or external stimuli. (In the case of automatic control the management system sets the conditions under which control is applied and the strength of response.)

14) Activation or deactivation of "direct routing and alternate routing".

15) Support for temporary re-routing schemes.

16) Management of the signalling network to ensure a consistent configuration of signalling resources.

17) Determination of attributes of signalling links including their functional status, error indications, traffic data or maximum bandwidth.

10 Addresses

Addresses are needed for various entities in the ASON control plane, as described below:

UNI Transport Resource: The UNI SNPP Link requires an address for the calling party call controller and network call controller to specify destinations. These addresses must be globally unique and are assigned by the ASON network. Multiple addresses may be assigned to the SNPP. This enables a calling/called party to associate different applications with specific addresses over a common link.

Network Call Control: The Network Call Controller requires an address for signalling.

Calling/Called party Call Control: The calling/called party call controller requires an address for signalling. This address is local to a given UNI and is known to both the calling/called party and network.

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

Routing Area: A routing area is given an address representing the collection of all SNPPs on that routing area, which is used for connection routing. It is unique within the scope of an administrative domain.

SNPP: An SNPP is given an address used for connection routing. The SNPP is part of the same address space and scope as subnetwork addresses.

Connection controller: A connection controller is given an address used for connection signalling. These addresses are unique within the scope of an administrative domain.

11 Connection availability enhancement techniques

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

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

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

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

The Protection or Restoration mechanism should:

– Be independent of, and support any, client type (e.g. IP, ATM, SDH, Ethernet).

– Provide scalability to accommodate a catastrophic failure in a server layer, such as a fiber cable cut, which impacts a large number client layer connections that need to be restored simultaneously and rapidly.

– Utilize a robust and efficient signalling mechanism, which remains functional even after a failure in the transport or signalling network.

– Not rely on functions which are non-time critical to initiate protection or restoration actions. Therefore consideration should be given to protection or restoration schemes that do not depend on fault localization.

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

Appendix I

ASON layer networks

The Automatic Switched Optical Network may be applied to layer networks. Examples of layer networks defined in other ITU-T Recommendations are provided in Table I.1. ASON may be also applied to other layer networks.

Table I.1/G.8080/Y.1304 – Layer networks: SDH, OTN and PDH

SDH		LOVC Path	VC-11
			VC-12
			VC-2
			VC-3
	HOVC Path	VC-4	
		VC-4-4c	
		VC-4-16c	
		VC-4-64c	
		VC-4-256c	
	Section	MSn, n=1,4,16,64,256	
		RSn, n=1,4,16,64,256	
		ES1 OSn, n=1,4,16,64,256	
		sSTM-1k, k=1,2,4,8,16	
		sSTM-2n, n=1,2,4	
		E31/P31s	
		E4/P4s	

OTN	Digital Path	ODU1
		ODU2
		ODU3
	Digital Section	OTUk, k=1,2,3
	Optical Path	OCh
	Section	OMSn
		OTSn
		OPSn

PDH	Path	P11x, P11s
		P12x, P12s
		P21x
		P22x, P22e
		P31x, P31e
		P32x, P32
		P4x, P4e
	Section	Eq, q=11,12,21,22,31,32,4

Appendix II

Bibliography

[B1] IETF RFC 2753: A Framework for Policy-based Admission Control, January 2000.

[B2] Unified Modelling Language (UML) (OMG UML Specification v. 1.3:OMG document ad/99-06-08).

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	**Y.1300–Y.1399**
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899

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

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	**Global information infrastructure and Internet protocol aspects**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2002