International Telecommunication Union

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

**G.7702**
(04/2022)

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

Data over Transport – Generic aspects – Transport network control aspects

---

# Architecture for SDN control of transport networks

Recommendation ITU-T G.7702

ITU-T

ITU-T G-SERIES RECOMMENDATIONS

**TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS**

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

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

# Recommendation ITU-T G.7702

## Architecture for SDN control of transport networks


### Summary

Recommendation ITU-T G.7702 describes the reference architecture for software defined networking (SDN) control of transport networks applicable to both connection-oriented circuit and/or packet transport networks. This architecture is described in terms of abstract components and interfaces that represent logical functions (abstract entities versus physical implementations).


### History

| Edition | Recommendation | Approval | Study Group | Unique ID* |
|---|---|---|---|---|
| 1.0 | ITU-T G.7702 | 2018-03-16 | 15 | 11.1002/1000/13540 |
| 2.0 | ITU-T G.7702 | 2022-04-06 | 15 | 11.1002/1000/14929 |


### Keywords

Application of SDN to transport networks, control plane interface (CPI), management-control system (MCS), MC components, transport SDN.


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

## FOREWORD

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

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

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

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

## NOTE

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

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

## INTELLECTUAL PROPERTY RIGHTS

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

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

## Table of Contents

Page

# Recommendation ITU-T G.7702

## Architecture for SDN control of transport networks

### 1 Scope

This Recommendation specifies the architecture and requirements for software defined networking (SDN) control of transport networks, consistent with the principles of SDN and complementary to SDN related work in ITU-T study groups 11, 13 and 17. This architecture is applicable to both connection-oriented circuit and packet transport networks.

The reference architecture describes SDN control of transport networks in terms of abstract MC components that are used for manipulating transport network resources in order to provide the desired functionality.

### 2 References

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


[ITU-T G.800] Recommendation ITU-T G.800 (2016), *Unified functional architecture of transport networks*.

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.

[ITU-T G.7701] Recommendation ITU-T G.7701 (2022), *Common control aspects*.

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

[ITU-T G.7711] Recommendation ITU-T G.7711/Y.1702 (2022), *Generic protocol-neutral management Information Model for Transport Resources*.

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

[ITU-T G.7714] Recommendation ITU-T G.7714/Y.1705 (2005), *Generalized automatic discovery for transport entities*.

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

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

## 3 Definitions

### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

**3.1.1** **adaptation** [ITU-T G.800].

**3.1.2** **address** [ITU-T G.7701].

**3.1.3** **administrative domain** [ITU-T G.7701].

**3.1.4** **architectural component** [ITU-T G.805].

**3.1.5** **call** [ITU-T G.7701].

**3.1.6** **call controller** [ITU-T G.7701].

**3.1.7** **calling/called party call controller** [ITU-T G.7701].

**3.1.8** **characteristic information (CI)** [ITU-T G.800].

**3.1.9** **client/server relationship** [ITU-T G.805].

**3.1.10** **component** [ITU-T G.7701].

**3.1.11** **component interface** [ITU-T G.7701].

**3.1.12** **connection** [ITU-T G.805].

**3.1.13** **connection controller (CC)** [ITU-T G.7701].

**3.1.14** **control domain** [ITU-T G.7701].

**3.1.15** **domain** [ITU-T G.7701].

**3.1.16** **layer network** [ITU-T G.805].

**3.1.17** **link** [ITU-T G.805].

**3.1.18** **link connection** [ITU-T G.805].

**3.1.19** **network call controller (NCC)** [ITU-T G.7701].

**3.1.20** **policy** [ITU-T G.7701].

**3.1.21** **recovery domain** [ITU-T G.7701].

**3.1.22** **resource database (RDB)** [ITU-T G.7701].

**3.1.23** **route** [ITU-T G.7701].

**3.1.24** **routing controller (RC)** [ITU-T G.7701].

**3.1.25** **routing domain** [ITU-T G.7701].

**3.1.26** **software defined networking** [ITU-T Y.3300].

**3.1.27** **subnetwork** [ITU-T G.805].

**3.1.28** **subnetwork connection** [ITU-T G.805].

**3.1.29** **subnetwork point (SNP)** [ITU-T G.7701].

**3.1.30** **subnetwork point pool (SNPP)** [ITU-T G.7701].

**3.1.31** **trail** [ITU-T G.805].

**3.1.32** **transitional link** [ITU-T G.800].

**3.1.33** **virtual network (VN)** [ITU-T G.7701].

## 3.2 Terms defined in this Recommendation

None.

# 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

| | |
|---|---|
| AAA | Authentication, Authorization and Accounting |
| APP | Application |
| AVC | Attribute Value Change |
| BRI | Boundary Resource Identifier |
| BSS | Business Support System |
| CC | Connection Controller |
| CCC | Calling/called party Call Controller |
| CCN | Control Communication Network |
| CI | Characteristic Information |
| CIM | Common Information Model |
| CPI | Control Plane Interface |
| DA | Discovery Agent |
| DS | Directory Service |
| EMS | Element Management System |
| FCAPS | Fault, Configuration, Accounting, Performance and Security |
| FP | Forwarding Point |
| IM | Information Model |
| LLDP | Link Layer Discovery Protocol |
| LRM | Link Resource Manager |
| MC | Management and Control |
| MCC | Management-Control Continuum |
| MCI | Management and Control Interface |
| MCS | Management and Control System |
| NCC | Network Call Controller |
| NMS | Network Management System |
| OSS | Operation Support System |
| PEP | Policy Enforcement Point |
| PM | Performance Monitoring |
| QoS | Quality of Service |
| RC | Routing Controller |
| RDB | Resource Database |
| SDN | Software Defined Networking |

| | |
|---|---|
| SLA | Service-Level Agreement |
| SNC | Subnetwork Connection |
| SNP | Subnetwork Point |
| SNPP | Subnetwork Point Pool |
| TAP | Termination and Adaptation Performer |
| TLS | Transport Layer Security |
| VN | Virtual Network |

## 5 Conventions

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

This Recommendation uses the diagrammatic conventions defined in [ITU-T G.7701] to describe controller components.

Within this Recommendation, the term layer is used as defined in [ITU-T G.800]. In this Recommendation the term level is used to describe the position in a hierarchy of SDN controllers or hierarchical transport network views.

## 6 Overview

An overview of the architecture of the control of transport networks is provided in clause 6 of [ITU-T G.7701].

This Recommendation defines the architecture for the use of SDN to control a transport network based on the SDN architecture described in [b-ONF TR-521]. This Recommendation defines the MC components and their interaction with the transport resources and management functions in other management control systems (MCSs). Figure 6-1 illustrates the relationship of SDN and other MCS to each other and to the resources in the transport network.

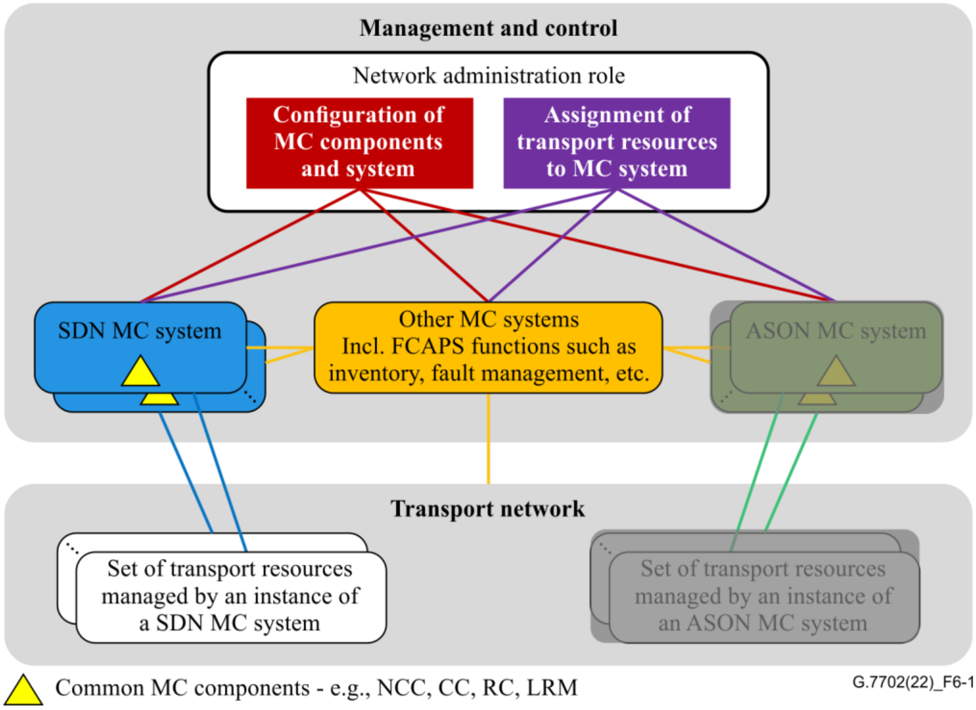


**Figure 6-1 – Management-control continuum for the transport Network showing SDN MC systems**

The SDN MC components are defined to manage a single transport layer network (including the media layer). A SDN controller includes the MC components required to manage one or more transport layer networks (see clause 7.5).

### 6.1 MC component overview

A reference architecture for the control plane that supports the requirements in this Recommendation appears in clause 6 of [ITU-T G.7701].

### 6.2 Call and connection control

Call connection and control are described in clause 6.1 of [ITU-T G.7701].

### 6.3 Functional characteristics of SDN for transport networks

Service providers may offer a wide range of services based upon differing business models. Operators continue to require protection of their commercial business operating practices and resources from external scrutiny or control, as well as to maintain the ability to differentiate the services they offer. The security and reliability of the underlying transport network remains a high priority. The SDN for transport architecture must thus provide for boundaries of policy and information sharing to accommodate, for example, the range of business models and varying trust relationships among users and providers, among users and among providers. Security at the boundary of SDN controller constrains the capabilities offered to clients across the control plane interface (CPI).

Transport network operators may select among a wide breadth of existing and emerging transport technologies, infrastructure granularity options, flexible capacity adjustment schemes, survivability strategies and infrastructure evolution choices. For a network operator/service provider, the optimal network layering, convergence choices and equipment selection depends upon multiple factors, such as:

– network size, geography, scalability
– service offerings portfolio, quality of service (QoS) committed in service-level agreements (SLAs)

– resource utilization, performance, survivability/resiliency trade-offs
– deployment schemes of control and management environment
– whether services traverse multiple operator domains

Consequently, the architecture for SDN control of transport networks must be designed to allow for multi-dimensional heterogeneity and not preclude network operators from optimizing their network design and supporting service realization as they see fit. The architecture must thus support the ability to decouple the services offered from their service delivery mechanisms and decouple QoS from its realization mechanisms.

As described in clause 11 of [ITU-T G.7701], distinct and independent sets of name spaces exist, from which identifiers are drawn, for:

– Resources in the transport network
– Control views of transport resources
– Management and Control (MC) components
– Control artifacts
– Reference points
– Control communications network (CCN).

These considerations lead to the following architectural principles:

– Provide a construct that reflects a service association that is distinct from its infrastructure/realization mechanisms
– Establish a modular architecture with interfaces at policy decision points (e.g., trust domain boundaries)
– Offer capability to distinguish identity from address (including distinguishing between and among, transport resources, MC components and control communication network addresses).

## 6.4 Architecture of SDN for transport networks

The architecture is described in terms of a hierarchy of SDN controllers that are inter-connected by a control plane interface (CPI) as shown in Figure 6-2. The level of a controller in a hierarchy is similar to, but distinct from, routing levels as defined in [ITU-T G.7701] and used in [b-ITU-T G.7715]. In both cases, level refers to a position within a hierarchy.

In SDN hierarchies, a server SDN controller presents a virtual network (VN) view of a sub-set of the transport network resources to its clients.

A SDN controller supports one or more clients, within this Recommendation, this is referred to as a client/server relationship. As shown in Figure 6-2 a SDN controller may have multiple clients and multiple servers; n+1 level controllers are clients of the level n controller, level n-1 controllers are servers (for the level n controller).

The client of a SDN controller may be either another SDN controller or a SDN application. Any SDN controller in the hierarchy may support a SDN application.

Within a SDN controller, a particular client is supported by a set of information including the transport network resources (presented in a VN), as well as the MC components required to support that client. Together, these are known as the client context.

The transport network resources managed by (in the scope of) a SDN controller are provided in one or more server contexts. Each server context is supported by a set of information relating to transport resources provided by a server as well as the associated MC components.

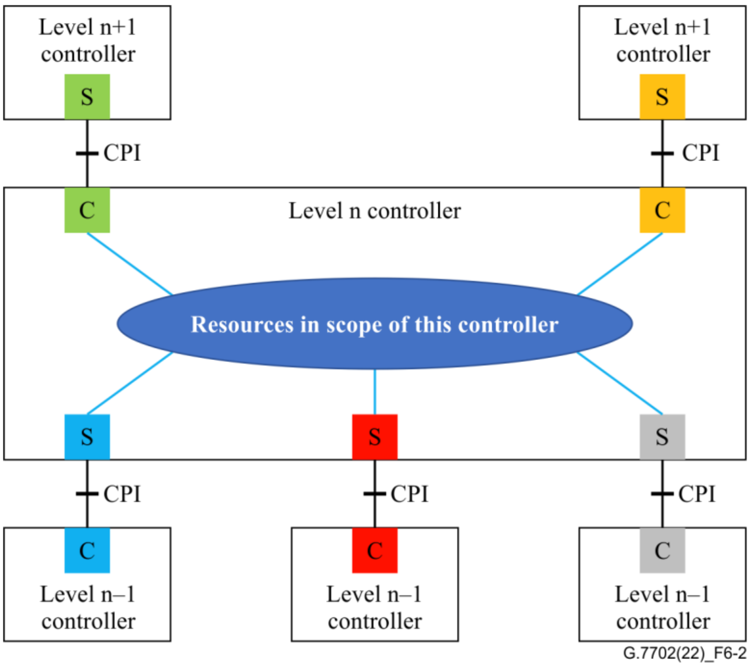


**Figure 6-2 – Client/server relationships in the SDN architecture**

### 6.4.1 Hierarchical SDN controller interconnection

Two SDN controllers that are adjacent in a hierarchy are interconnected via one pair of client and server contexts. The SDN controller with the client role has a server context with a 1:1 relationship with the client context in the SDN controller with the server role. The server SDN controller presents a virtualized view of the transport resources to the client SDN controller using subnetwork point (SNP) and subnetwork point pool (SNPP) identifiers. The client SDN controller only has SNP/SNPP identifiers for resources in its scope and therefore cannot configure forwarding for those resources. In this case, the client controller sends a request to the server controller. The case where a SDN controller has visibility of the ITU-T G.800 forwarding point (FP) name space is described in clause 6.5.

Figure 6-3 shows the hierarchical architecture of SDN for transport networks.

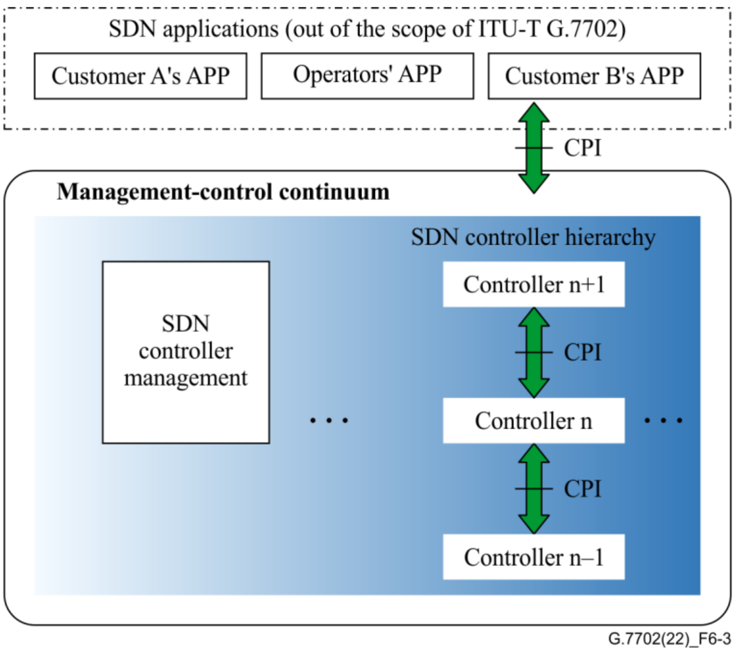


**Figure 6-3 – Hierarchical architecture of SDN for transport networks**

### 6.4.2 Peer-to-peer SDN controller interconnection

In some cases, SDN controllers may be deployed as peers instead of in, or in addition to deployment in, a hierarchy as described above. To support peer-to-peer interactions the client and server contexts are connected a shown in Figure 6-4.

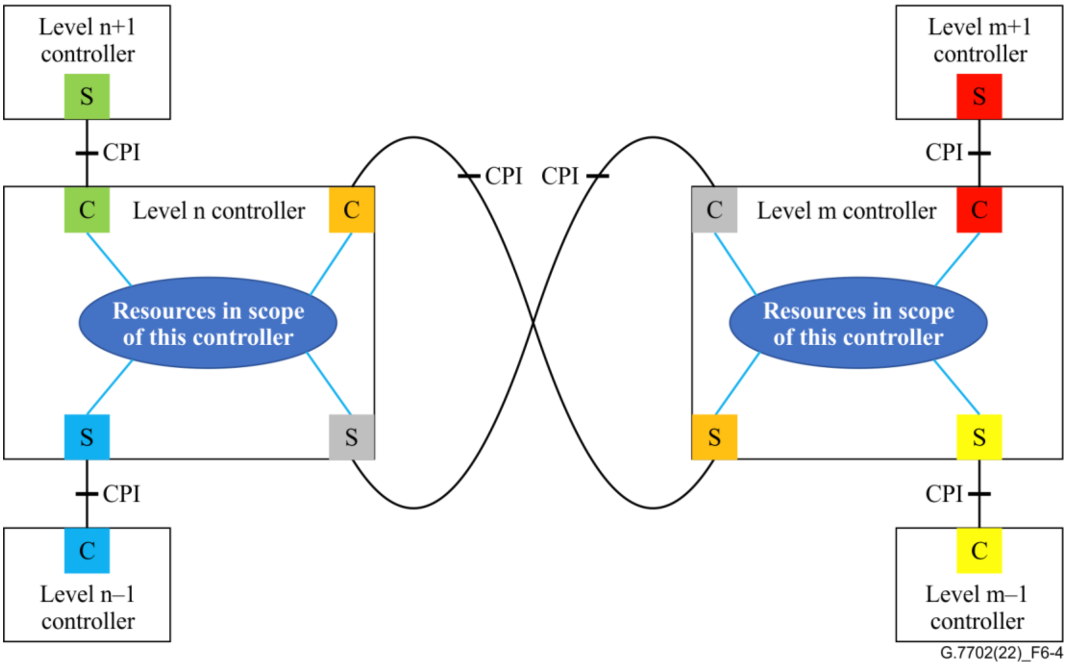


**Figure 6-4 – Peer-to-peer interconnection of SDN controllers**

In a hierarchical arrangement the level n controller (acting as a server) provides resources to the level n+1 controller (acting as a client). In a peer-to-peer relationship (shown in Figure 6-4) the level n controller (acting as a server) provides resources to the level m controller (acting as a client) in the peer, similarly the level m controller (acting as a server) provides resources to the level n controller in its peer (acting as a client). Note that in this example of peer-to-peer interconnection we use 'n' and 'm' to indicate that the peer controllers may be at arbitrary positions in their respective hierarchies.

A SDN controller that is participating in a peer-to-peer relationship can also participate in a hierarchal relationship with other SDN controllers.

## 6.5 Interaction between SDN controller and transport network

When MC components in a SDN controller have visibility of the [ITU-T G.800] forwarding point (FP) name space that SDN controller is able to directly control those transport resources. Transport resources for which the controller has visibility of the FP name space are referred to as local resources and are placed in a server context that does not provide an external interface That is, the CPI is not exposed.

A SDN controller at any level in the hierarchy may have local resources in scope. A SDN controller may have both local resources and resources provided by one or more server SDN controllers in scope. This is shown in Figure 6-5.

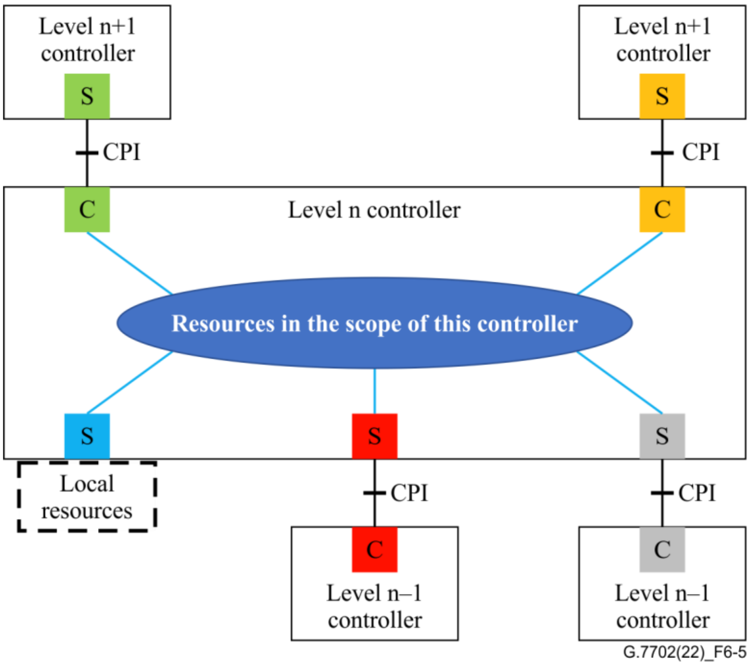


**Figure 6-5 – SDN controller with local resources in scope**

## 6.6 Interaction between SDN controller and applications

The client context in a SDN controller at any level in the hierarchy can communicate with the server context of another SDN controller as described in clause 6.4 or it may communicate with an application.

Applications can be supported by SDN controllers at any level in the hierarchy, the server SDN controller cannot distinguish between a client that is an applications and client that is a SDN controller. The interaction between SDN controller and an application is also through CPI as shown in Figure 6-3, with the appropriate information exchange and security related configurations.

## 6.7 Management functions interactions

The management-control continuum described in [ITU-T G.7701] expresses the view that management and control functions are essentially the same and thus they can be grouped into one set of management and control (MC) functions. There are MC functions that directly manage resources and MC functions that manage other MC functions (e.g., configuration of call control). Many of these MC functions are still essential in a SDN environment. Relevant management functions include those for supporting fault management, configuration management, accounting management, performance management and security management (FCAPS) as described in [b-ITU-T M.3400]. For example, in the transport network, management is minimally required for initial configuration of the transport network resources, assigning the SDN-controlled parts and configuring their associated SDN controller(s). In the SDN controllers, management needs to configure the policies defining the scope of control given to the controller's client and to monitor the performance of the system. Management configures the security associations that allow functions to safely communicate among each other. Additional examples of management functionalities include equipment inventory, software upgrade, fault isolation, performance optimization, energy efficient operations and autonomic management (continuous adaptation to the network status).

Both SDN controllers and OSSs/NMSs may contain MC functions that provide transport network service and resource management functions. In the application of SDN to transport network, the controller's main function is to provide connection and routing control related management functions. NMS/element management systems (EMS) typically contains transport network and elements management functions known as FCAPS.

The FCAPS for an SDN controller may also be contained within the SDN controller management function, as illustrated in Figure 6-1.

Incorporating FCAPS into the SDN architecture is possible by considering the name/identifier spaces that MC functions use. [ITU-T G.7701] components use boundary resource identifier and SNPP name spaces and the discovery agent (DA) and termination and adaptation performer (TAP) components also use one of the resource name spaces. FCAPS functions use resource name spaces as well. This enables both ITU-T G.7701 components and FCAPS functions to be accessed in the management-control continuum.

# 7 Transport resources and their representation

## 7.1 Transport resources and their representation

The transport functional architecture is described in clause 7.1 of [ITU-T G.7701]

## 7.2 Domains

Domains are described in clause 7.2 of [ITU-T G.7701]. The CPI is used at the boundary between SDN control domains.

## 7.3 Control view of transport resource for connection management

The control view of transport resources is described in clause 7.3 of [ITU-T G.7701].

## 7.4 Virtualization

Abstraction and virtualization of resources are described in clause 7.4 of [ITU-T G.7701]

The VN represents a part of the network resources information contained in a client context. Transport network resources are assigned to a VN in a client context by administrative or other means. The network resources information in the server context of a client controller is the same as the network resources information in the VN of the server controller. The use of the common information model (CIM) to represent the network resources are provided in Annex A.

Figure 7-1 illustrates the basic method by which a server controller realizes transport resource virtualization to provide a VN for each client controller.

- The transport resources in the scope of the server controller are mapped from the controller's server contexts (server context specific name space and identifiers) into the controller's local resource pool (common name space and identifiers) in the resource database (RDB).
- A subset of the transport resources in the controller's local resource pool are assigned to the three client contexts.
- The resources allocated to each client context are then mapped to the client context name space and virtualized to provide a VN for each client controller.

The client context in the server controller and the server context in the client controller must use the same name space. The name spaces used in the server controller's local resource pool and client controller's local resource pool may be different. One single name space may be used if the hierarchical controllers are in a trusted relationship (e.g., both within the same administrative domain), and in this case the name space mapping to/from the local resource pool is not required.

Interlayer client support is described in clause 7.6 of [ITU-T G.7701].

The internal function view of controller is illustrated in Figure 7-2.

- The server context is directly created and configured by the controller admin to receive the network resource information from the client context of the lower level controller or physical network. The view translation is needed to normalize the way that the lower level network resources information is presented. The view translation may include name space translation, network resource refactoring and network resources state mapping. Name space translation refactoring and network resource state mapping may not be needed in a trusted relationship. The network resource information is stored in the controller's local resource pool of RDB with local name space. The inverse direction actions will be carried out if the network resources manipulation requests are triggered by the client contexts.

- Function of link discovery and management between server contexts is needed to discover the link connections between server contexts. The link discovery process between different administrative domains is usually configured manually. The link connection information between server contexts is a part of the common resource pool and is stored in the controller's local resource pool of RDB to construct the topology.
- The client contexts are created as a result of the requests from the external clients. According to the detailed resource requirements from the client, the controller admin allocates the resources to the client context by allocating the required resources from the controller RDB. The resources allocated to the client context could be the normalized full detail of the resource information held by the controller or a subset of the resource information. The representation of the resources may be refactored, abstracted and virtualized before being made visible to the client. The state of the resources presented to the client must be derived by a mapping from the state of the local network resources. The use of name space translation, virtualization and abstraction allows the server to hide details of the underlying network from the client. The inverse direction actions will be carried out if the network operation commands are received from the above clients.
- There may exist more than one name space in one client context, since in the untrusted relationship, the name space of local network resources may be different from that in the upper level controller.
- The configured connection information should also be stored in the controller RDB, since these connections represent the network resources that have been consumed and this may impact the resources available to other client contexts.
- There may exist multiple users for one client. Correspondingly, the client context should be able to provide different views for these users. Each view could enforce the policy decisions at the policy enforcement point (PEP).

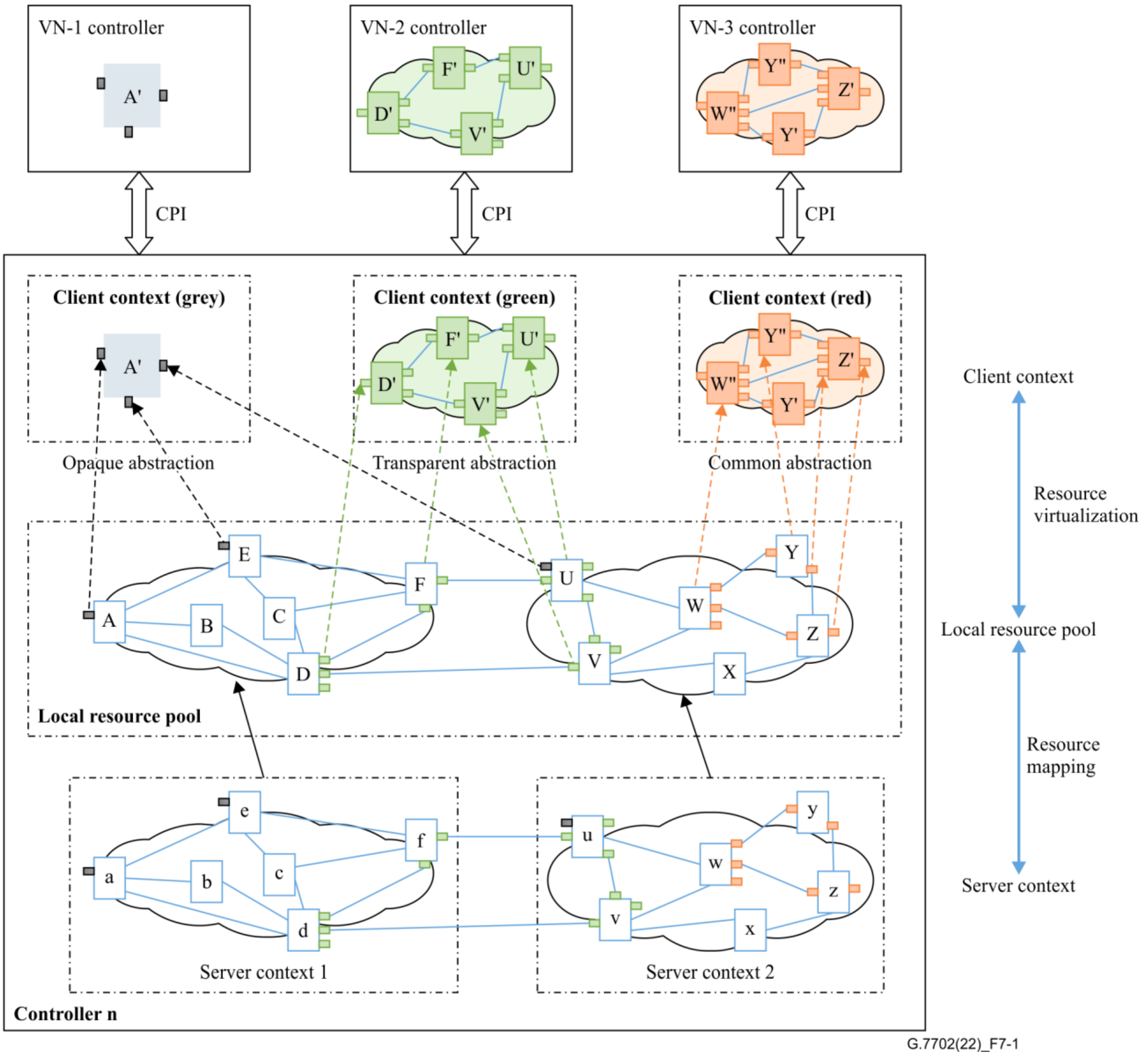


**Figure 7-1 – Example of transport resources virtualization**

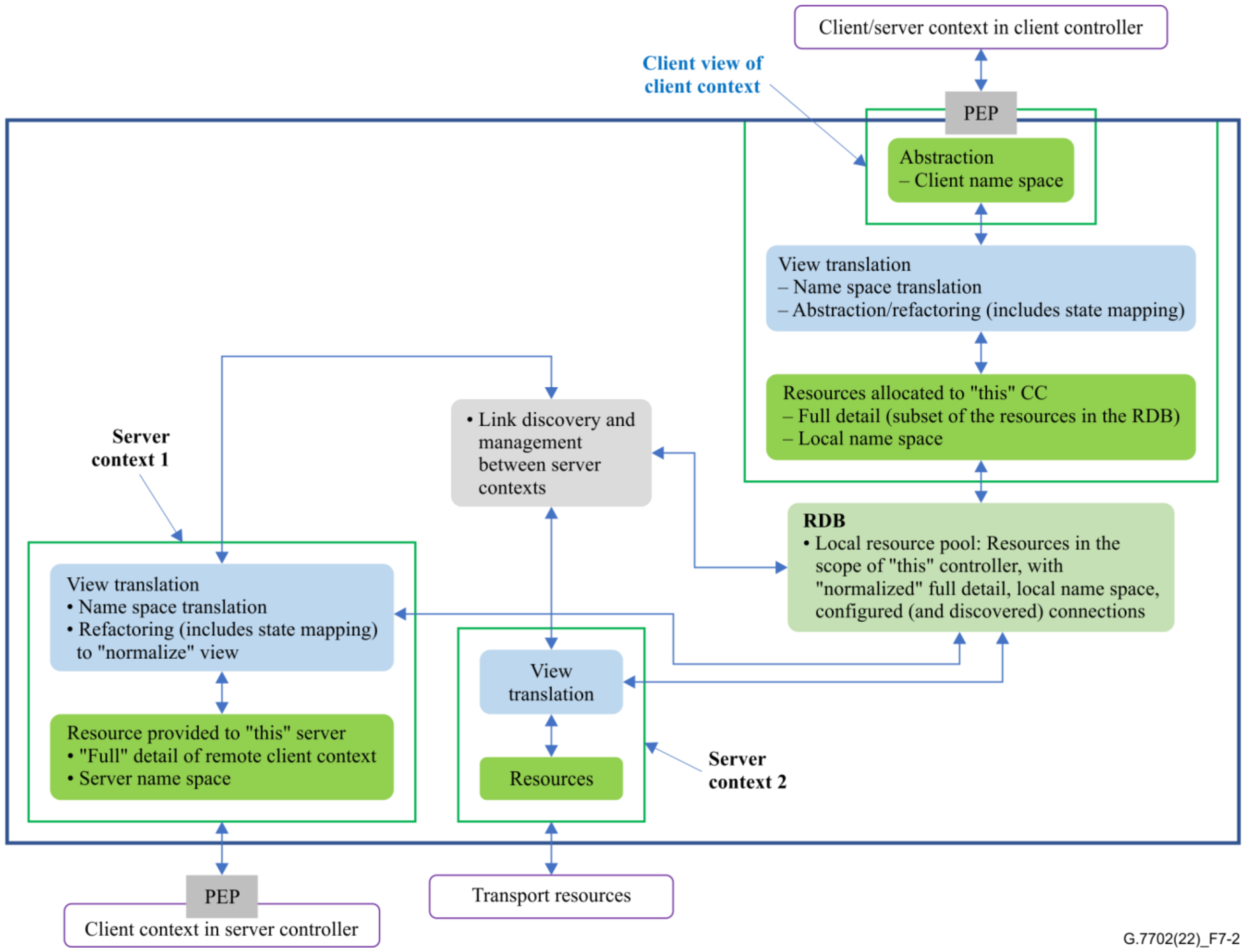


**Figure 7-2 – View of the internal functions of a controller**

## 7.5 Multi-layer control

Multi-layer aspects are described in clause 7.5 of [ITU-T G.7701].

MC functions are described in the context of the management of a single transport layer network. To control a multi-layer transport network single layer network SDN controllers are arranged in a client/server level hierarchy that corresponds to the client/server layer network relationships in the transport layer networks being controlled. The server controller provides a virtualized view of the topology and connectivity of the underlying server layer network. The server controller manages the client (layer network) to server (layer network) adaptation and presents the resources to the client controller as a client layer SNPP link, or as client layer subnetworks interconnected by SNPP links. If the resources available to the client layer network are insufficient to support a connection request, additional resources may be provided by activating or creating new connections in one or more server layer networks. Operator policies govern the availability of underlying server layer resources to the client layer. More details on the use of SNP link connections and of SNPP links containing multiple SNP links are provided in clause 7.5.1 and clause 7.5.2 of [ITU-T G.7701].

The communication between the controllers in adjacent levels is supported by a (multi-layer) network call controller (NCC). This is illustrated in Figure 7-3.

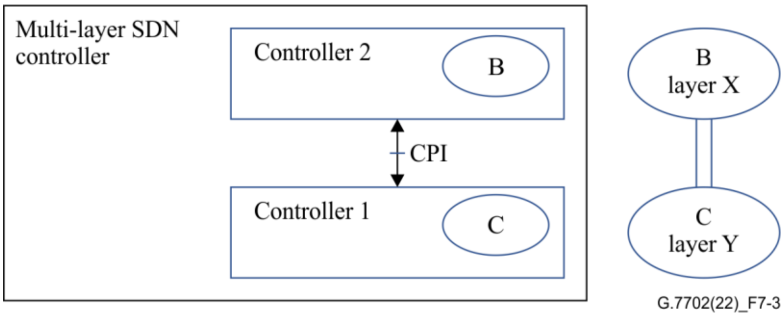


**Figure 7-3 – Multi-layer network control**

A multi-layer SDN controller has resources from more than one transport layer network in scope. A multi-layer SDN controller can be described as a set of single layer SDN controllers each managing a single layer network. In a multi-layer SDN controller the CPI between the (single layer) SDN controllers is not exposed, and the resource database (RDB has all of the (multi-layer) resources in scope and includes transitional links between the layer networks to support multi-layer path computation. The results of a multi-layer path computation are passed to the appropriate MC functions for each of the layer networks. The multi-layer controller ensures that the connections in each layer network are added, modified or deleted in the appropriate sequence.

### 7.6 Interlayer client support

Interlayer client support is described in clause 7.6 of [ITU-T G.7701]. To achieve this with SDN MC systems, NCCs in hierarchically arranged SDN controllers would use their interlayer interfaces.

### 7.7 Calls supported by calls at same layer

Calls supported by calls in the same layer network are described in clause 7.7 of [ITU-T G.7701]. To achieve this with SDN MC systems, several SDN controllers can be used to perform concatenation of calls/connections over the resources each controller spans. The type 3 interaction of clause 15.3 is used.

### 7.8 Mapped server interlayer relationships

Mapped server interlayer relationships are described in clause 7.8 of [ITU-T G.7701]. For SDN MC systems, hierarchically arranged SDN controllers can be used to perform 1:1, 1:n and n:1 mapped server relationships. The type 5 interaction of clause 15.5 is used.

## 8 MC components

This clause describes the MC components of the SDN controller using the approach described in [ITU-T G.7701]. These components are used to represent the MC functions within an SDN controller.

### 8.1 Notation

The notation used is described in clause 8.1 of [ITU-T G.7701]

### 8.2 Policy

The use of policy clause is described in clause 8.2 of [ITU-T G.7701].

### 8.3 Common components

The use of the MC components is described in clause 8.3 of [ITU-T G.7701].

#### 8.3.1 Call controller components

The network call controller is described in clause 8.3.1 of [ITU-T G.7701].

In [ITU-T G.7701], the call controller is a component with important policy boundary functions for entities that are in a relationship. The call may have associated connections and this state is also maintained by the call controller.

The interfaces described in [ITU-T G.7701] are extended for use in SDN as shown in the tables below.

The network call controller component interface descriptions in Table 8-3 of [ITU-T G.7701] are augmented by the addition of the 'Server NCC coordination in' and 'Connection request out' interfaces shown in Table 8-1.

**Table 8-1 – Network call controller component interfaces**

| Input interface | Basic input parameters | Basic return parameters |
|---|---|---|
| Server NCC coordination in | A pair of local SNP identifiers and optionally a route.<br>An optional ordered list of SNP pairs. | A subnetwork connection (controller response with the established connection) |
| **Output interface** | **Basic output parameters** | **Basic return parameters** |
| Connection request out | Boundary resource identifier,<br>Ordered list of SNP pairs | An ordered list of pairs of SNPs |

### 8.3.2 Connection controller (CC) component

The common description for the CC component is provided in clause 8.3.2 of [ITU-T G.7701].

In an SDN controller the CC establishes and releases connections on VNs within a client context. Information the CC may need about links such as utilization, is maintained in the resource database (RDB). If an ordered list of SNP pairs is supplied by an NCC when requesting a connection, these are used in the request to the routing controller (RC) for resolution of the route. Connections established and their associated identifiers, are CC artifacts that are also maintained in the RDB.

For the 1: n case of SDN controllers that use the NCC to NCC interfaces, the CC's behaviour is similar to that in the hierarchical routing case. The single client NCC makes multiple NCC calls to the 'n' servers and gathers a pair of SNPs per call. Each SNP pair represents the returned connection. At the client NCC, the pairs need to be concatenated and it is the connection controller (CC) that does this. The Connection request out interface from the NCC described in clause 8.3.1 includes an ordered list of subnetwork connection (SNC) pairs that the client CC uses when constructing the connection.

### 8.3.3 Routing controller (RC) component

The common description for the RC component is provided in clause 8.3.3 of [ITU-T G.7701].

In an SDN controller, the RC component computes routes based on the resources in the client's VN. It maintains topology information that is logically in the RDB and the relationship between a VN and its underlying topologies in server contexts.

### 8.3.4 Link resource manager (LRM) component

The common description for the link resource manager (LRM) is provided in clause 8.3.4 of [ITU-T G.7701].

In [ITU-T G.7703] and [ITU-T G.7701] a different LRM is responsible for each end (LRMA, LRMZ) of an SNPP link. A logically centralized controller found in SDN has a global view of resources in the network. As a result, the LRM in an SDN controller covers both A and Z scope of SNPP links in the SDN control domain. For those SNPP links between SDN control domains, the LRM only covers the A or Z end in scope.

### 8.3.5 Discovery agent (DA) component

The common description for the discovery agent (DA) component is provided in clause 8.3.5 of [ITU-T G.7701].

### 8.3.6 Termination and adaptation performer (TAP) component

The common description for the termination and adaptation performer (TAP) component is provided in clause 8.3.6 of [ITU-T G.7701].

### 8.3.7 Directory service (DS) component

The common description for the directory service (DS) component is provided in clause 8.3.7 of [ITU-T G.7701].

The DS component may also be used by other components when mapping between local topology and underlying topology information in server contexts.

### 8.3.8 Notification component

The common description for the notification component is provided in clause 8.3.8 of [ITU-T G.7701].

A transport resource (e.g., a link) may contribute to many VNs and some of the contexts associated with those VNs may subscribe to a type of notification regarding that resource (e.g., an alarm). When a change occurs that is cause for notification, the responsible MC component in each of the affected contexts recognizes the change and generates the appropriate notification(s).

### 8.3.9 Protocol controller (PC) component

The common description for the protocol controller component appears in clause 8.3.9 of [ITU-T G.7701].

The protocol controller maps the parameters of the abstract interfaces of the MC components into messages that are carried by a protocol to support interconnection via an interface.

### 8.3.10 Traffic policing (TP) component

The common description for the traffic policing component is provided in clause 8.3.10 of [ITU-T G.7701].

This traffic policing component's role is to check that the incoming user connection is sending traffic according to the parameters agreed upon.

## 9 Control communications network

The application of SDN to transport network requires a control communications network (CCN) to transfer information e.g., between SDN controllers at different levels, between SDN controller and the resources in their scope, or between SDN controllers and their management functions. The CCN is supported by the DCN functions specified in [ITU-T G.7712].

The CPI and management and control interface (MCI) are the interfaces defined for these information transfers; the information transfers themselves require a CCN between the communicating entities.

The information transported via different instances of the CPI and MCI varies e.g., it may relate to connectivity, compute requirements, dimensioning, reliability, performance, security, etc.

The reliability and security required from the CCN may vary depending on particular usage. Appropriate implementation mechanisms should be used to support reliable information transfer in the CCN if that is required. The CCN should provide appropriate security in terms of access control and guaranteeing secure transport of information. Mechanisms such as authentication, authorization and accounting (AAA) and transport layer security (TLS) may be appropriate solutions to those challenges.

Several different communication scenarios and how some aspects of the communication vary between them are examined below:

**Control communication network between controller and transport resources:**

– The information carried over the interface between controller and resources in its scope is used for topology discovery, path computation, connectivity control and maintenance purposes.
– Controllers establish resource topology by processing adjacency information provided by those resources.
– A controller processes a connection request from one of its clients by establishing the required path and then sending configuration commands to the appropriate resources on that path.
– Resources can communicate changes in their state to the controller, which, for example, provides the information for network or service monitoring applications.

**Control communication network between controllers at different levels:**

– Services are established across a network by the appropriate and coordinated configuration of the resources in the scope of one or more controllers.
– Information is communicated between controllers to coordinate this resource configuration. For example, signalling messages exchanged between NCCs and the routing information exchanged between RCs, are transported over the interface between controllers at different levels allowing the coordination of network configuration that establishes service for the client.

**Control communication network between SDN controllers and management applications:**

The information transferred over the interface in this scenario is related to fault, configuration and performance management of the controller. Management applications are concerned with the control, surveillance and performance of the controller.

A controller which suffers from fault may be replaced with the auxiliary one according to the configuration of management functions once the fault is detected.

Performance information from the controller, such as congestion control should be reported to or can be acquired by the management application through the control communication network.

## 10 Management aspects

This clause deals only with management of SDN controllers themselves. Management of resources controlled by the controller and management of SDN applications themselves are out of scope of this Recommendation. In general, the customer of a transport network operator may interact with the network using an SDN application. It is a local matter for such applications to deal with details of customer management (authorization, billing, reporting, etc.). The application may use information obtained from its interaction with the SDN controller e.g., when it invoked a call setup and for whom, to generate billing and reporting details, etc.

### 10.1 Management of SDN controllers

The SDN controllers should support configuration management, fault management, performance management and security management for the SDN controllers, including:

– The initial parameters configuration for controllers, e.g., the address, name ID, client-server relationships, enable and disable, etc.

– The status management of controllers themselves, including the process status of software and hardware, the usage status of controllers' resources, operating and running status and management status.

– The alarms and events management of controllers, e.g., the node failure, the modules or components failure, the process failure of software and hardware, etc.

– The performance management of controllers, e.g., the configuration of reports for performance monitoring, the performance of controlling process, the running performance of controllers' resources, etc.

– The policy management for controllers, including the mapping policy for quality of service (QoS), the restriction policy for routing, the security policy, the policy of protection and restoration, etc.

### 10.2 Management of control plane interfaces (CPI)

The management aspects of control plane interfaces include the management of CPI illustrated in Figure 6-2, which include:

– The type of interface, the type of interface protocol, address, identifier, etc.

– The type of SDN control signalling protocol and related parameters, etc.

– Access control and related policy management of CPI, including access user management, e.g., creation access username and password, access authority, access security policy, etc.

### 10.3 Management of control communication network (CCN)

The management aspects of CCN include:

– The CCN configuration management, including CCN channels, interface address and identifier, transport mode, protection and restoration, etc.

– The status monitoring of CCN, including alarms (e.g., the communication failure between controllers or other planes), performance, etc.

## 11 Identifiers

A number of distinct and independent sets of name spaces, from which identifiers are drawn are described in clause 11 of [ITU-T G.7701]:

– Resources in the transport network;

– Control view of transport resources;

– MC components;

– Control artifacts;

– Reference points;

– Control communications network.

Each of these name spaces and the identifiers that are drawn from them is described in clause 11 of [ITU-T G.7701]. Further information on the use of these name spaces and identifiers in the context of SDN is provided in the remainder of this clause.

### 11.1 Resources in the transport network

The identifiers used for the resources in the transport network are described in clause 11.1 of [ITU-T G.7701].

The only SDN components that use these identifiers are the DA and TAP.

## 11.2 Control view of transport resources

The name spaces used to provide a control view of transport resources are described in clause 11.2 of [ITU-T G.7701].

## 11.3 MC components

The name spaces used for MC components are described in clause 11.3 of [ITU-T G.7701].

## 11.4 Control artifacts

As described in clause 11.4 of [ITU-T G.7701], the MC components create and use control artifacts including, for example, connections, routes and calls. Normally the control component that creates a control artifact assigns an identifier. These identifiers are drawn from an independent name space.

## 11.5 Reference points

Reference points are described in clause 11.5 of [ITU-T G.7701]. The following subclauses describe how these reference points are used in SDN.

### 11.5.1 SDN control plane interfaces

The SDN control plane interfaces with other entities are shown in Figure 11-1. These interfaces are reference points for information hiding, traffic and namespace isolation and policy enforcement. Reference points of MC components could be bundled in these interfaces.

The control plane interfaces include:

a) The relationships between transport NEs and SDN controllers, between SDN controllers at adjacent levels in a recursive hierarchy (see clause 10) and between SDN controller and applications, are all client and server. North-south instances of these same interfaces with the same set of information models (IMs) are used between each level of the hierarchy and may represent (see clause 7) virtual resources.

b) In the management-control continuum (MCC), an administration role acts to configure and manage SDN controllers. The interface between a SDN controller and the administration role is an MCI – see Figure 11-1. Note that the administration role may be implemented in a legacy management system (e.g., EMS/OSS/BSS).

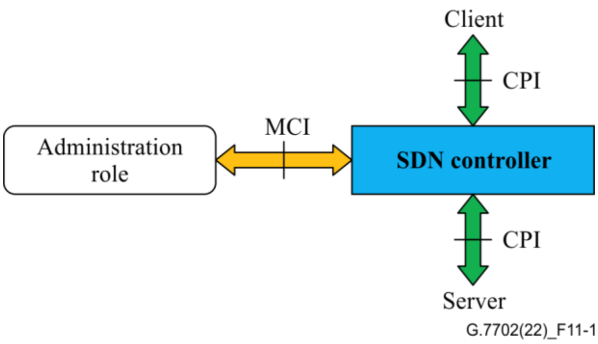


**Figure 11-1 – General architecture of SDN controller interfaces**

Figure 11-2 shows a 1:1 hierarchical stack of controllers at different levels. The control function inside the transport NE could be seen as a management and control agent of SDN in the NE.

Note that in Figure 11-2 the N+2 level controller could be an SDN application; the server SDN controller at level N+1 is, however, only aware of the presence of a client, i.e., it cannot distinguish between an application and another SDN controller.

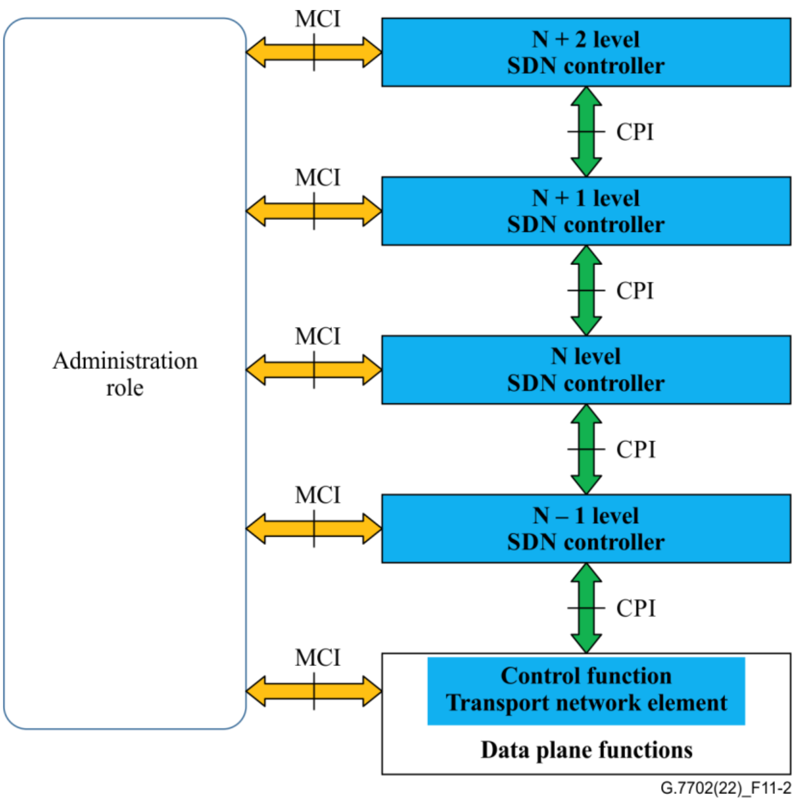


**Figure 11-2 – 1:1 Hierarchical stack of controllers**

Figure 11-3 shows an m:1 hierarchical arrangement of SDN controllers where the server has created separated client contexts for each of its three clients and assigned resources to each of those clients (the resources are visible in the context). This is referred to as 'resource partitioning'. Note that, although the clients of the low level controller are all shown here as SDN controllers, the low level controller can also support applications over the CPI. In general, a controller is unaware of the role or naming convention applied to its CPI clients (see clause 6.4.1). By extension this means that applications can appear as clients of a controller at any level in a hierarchical arrangement of controllers.

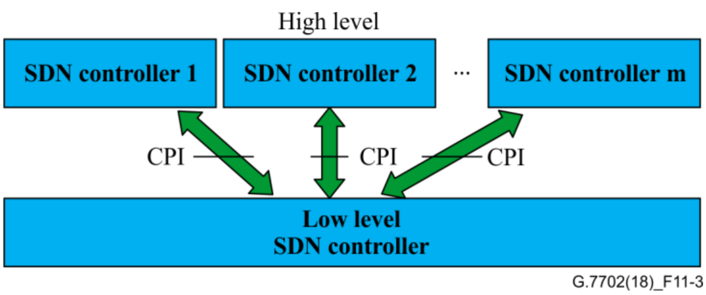


**Figure 11-3 – m:1 hierarchical arrangement of controllers in partitioned view**

Figure 11-4 shows a 1:n hierarchical arrangement of controllers where the high level controller has established relationships with three server controllers each of which provide it resources. The entire set of resources available to the high level controller are referred to as a 'virtualized view'.

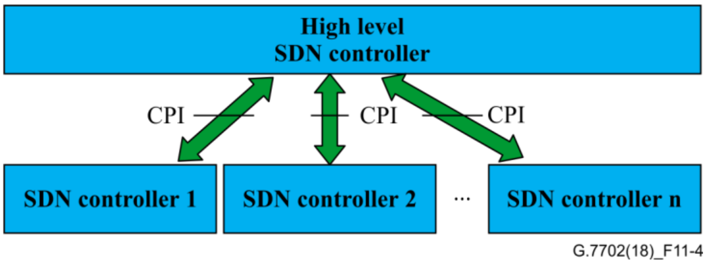


**Figure 11-4 – 1:n hierarchical arrangement of controller in virtualized view**

To migrate a transport network to SDN control, one approach is to upgrade the EMS/NMS with the SDN controller function (such as call control and etc.) and a CPI agent, which allows interaction with a higher level SDN controller or application as illustrated in Figure 11-5. The EMS/NMS enhanced with SDN controller functionality could reuse a proprietary interface, or other protocol (e.g., Netconf/YANG) to communicate with the legacy transport network elements which can remain unchanged.

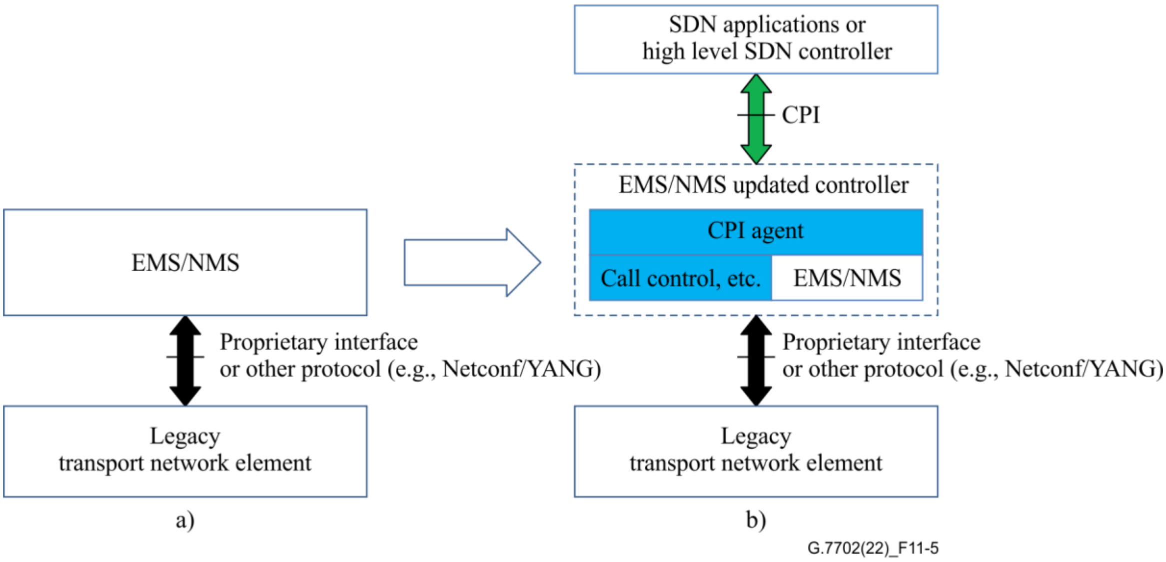


**Figure 11-5 – A migration approach to support SDN control for a legacy transport network**

A boundary resource identifier (BRI) is used to identify both the transport resource(s) and the interface(s) to MC components at the boundary of a domain. The BRI is drawn from an independent name space. The use of an independent name space avoids exposing the identifiers that the MC components use within the domain.

The CPI between the two SDN controllers shown in Figure 7-3 is an example of an interface that could be identified by a BRI.

### 11.5.2 Functional requirements for identifiers in the CPI

The CPI should support several categories of functional requirements, including topology, connection, path computation and resource virtualization/abstraction, performance and fault management, notification/events, etc.

#### 11.5.2.1 Topology management functions

The CPI should support the topology management function and retrieval of topological information, which should include:

– Topology management for intra-domain and multi-layer transport networks;

– Topology management for inter-domain networks or links;
– Abstraction of topology and resources of transport networks.
– Getting detailed topological information;
– Receiving topology updates from SDN controllers;
– Retrieval of logical network topologies that include combinations of logical and physical topologies.

#### 11.5.2.2 Connection control function

Through a CPI, SDN controllers should support connection control functions including connection creation, connection modification, connection deletion, OAM configuration and activation and survivability mechanisms.

#### 11.5.2.3 Instantiation/Deletion /Update of services

Controllers are responsible for computing a path for a connection request and providing the path to their client controllers or applications through a CPI, which should support:
– Path computation of intra-domain, inter-domain or multi-layer for the transport network, according to the path constraint conditions in the connection request;
– Optimization of P2P paths, reconfiguring paths to achieve optimization;
– Requests for path (route) information needed to set up connections.

#### 11.5.2.4 Resource virtualization/abstraction

A lower level controller can present its transport resource and virtual networks view to its client controllers and applications through the CPI. Client controller interaction over the CPI allows:
– VN creation, modification and release request.
– Partition of network resources for different VNs according to policy.
– Providing different views (abstractions) of the resources, real and virtualized, under its control, subnetwork and link topological resource can be represented with varying degrees of detail (granularity).

#### 11.5.2.5 Performance and fault management

Lower level controllers could provide performance and fault management related information to their clients through the CPI. Performance monitoring and fault detection are necessary for transport resources and virtual network resources. Dynamic service control policy enables the performance and fault management function.

#### 11.5.2.6 Notifications and events

Notifications refer to the set of autonomous messages that provide information about events, for example, alarms, performance monitoring (PM) threshold crossings, object creation/deletion, attribute value change (AVC), state change, etc., related to resources in the controller's scope. These notifications may also include messages indicating service faults, performance degradation and so forth.

Notifications are provided over the CPI by server controllers to their client. The mechanism used is a matter of implementation detail.

#### 11.5.2.7 MC component information flows

Some of the information listed above that are supported over the CPI include the following MC component information flows: NCC-NCC and RC-RC.

### 11.5.3 Interaction between NCCs over CPI

In Figure 15-1, the NCC in the client controller sends the call request to the NCC in the server controller. The resource identifiers (i.e., SNP identifiers/name spaces) are sent from client NCC to server NCC.

Through a CPI, a client NCC can initiate connection creation, connection modification, connection deletion and connection query requests. The interactions between NCCs over CPI are listed in this clause.

#### 11.5.3.1 Call creation

If the connection is setup is successful, the client NCC obtains connection identifiers and connections.

The parameters sent from the client NCC to the server NCC over CPI could be:

– SNP identifiers (or Name) in the client context;
– Direction of connection;
– Connection constraints including capacity, layer, latency, cost, etc.
– Required capacity.

The resulting parameters returned from server NCC to client NCC over CPI could be:

– Success/Failure;
– Connection identifiers in server context;
– Operational state;
– Connection details, see above parameters.

#### 11.5.3.2 Call modification

The parameters sent from the client NCC to the server NCC over CPI could be:

– Connection identifiers in client context;
– Connection constraints including capacity, layer, latency, cost, etc.
– Required capacity.

The resulting parameters returned from the server NCC to the client NCC over CPI could be:

– Success/Failure;
– Operational state;

#### 11.5.3.3 Call deletion

The parameters sent from the client NCC to the server NCC over CPI could be:

– Connection identifiers or name;

The resulting parameters returned from the server NCC to the client NCC over CPI could be:

– Identifiers of the deleted connection;
– State changes.

#### 11.5.3.4 Call query

The parameters sent from the client NCC to the server NCC over CPI could be:

– Connection identifiers in client context;
– Connection constraints including capacity, layer, latency, cost, etc.

The resulting parameters returned from the server NCC to the client NCC over CPI could be:

– Connection identifiers in the server context;

– Connection details.

### 11.6 Control communications network

The control communications network (CCN) that allows the MC components to exchange information is described in clause 11.6 of [ITU-T G.7701].

An MC function operates on resources using a resource view. This view is enabled by a name/address/identifier plan. For example, the [ITU-T G.7701] routing controller component views topology using the SNPP name space and the TAP component manages the mapping of the SNP name space to the FP name space (resource label).

For the purposes of some MC functions, there does not have to be a distinction between VN and non-VN resource views. If the resource is a networking resource, path computation can operate on a combined topology without needing to know the difference.

An SDN controller may have MC functions that have resources in scope whose FP name space is visible, as well as resources in scope whose FP name space is invisible. Note that the routing function only uses the SNP/SNPP name space, so it is not affected by FP name space's visibility. For discovery, TAP and LRM, the visibility of FP name space becomes important.

MC functions in an SDN controller that configure FPs associated with transport network resources do so via the resource-control interfaces [ITU-T Y.3300] and the configuration and/or properties exposed are abstracted by means of information models (IMs) [ITU-T G.7711] and data models (e.g., Yang model). Specifically, the resource-control interface is the generic interface across which an instance of the SDN information model is managed. This interface provides high-level accesses to the network resources regardless of their respective technology. As the SDN architecture operates on an abstract model of the transport resources, there is no architectural distinction between the control of physical and virtual transport NEs. Each may have FP name spaces whose configuration enables transfer of information.

## 12 Resilience

Resilience for MC components is described in clause 12 of [ITU-T G.7701]

## 13 Connection availability enhancement techniques

Connection availability enhancement techniques are described in clause 13 of [ITU-T G.7701]. The role of the SDN RC, CC and NCC in connection availability enhancement is illustrated in Figure 13-1 of [ITU-T G.7701].

## 14 Topology and discovery

In the logically centralized and hierarchical control mode of SDN, the topology of the transport network is maintained by the co-operation of multi-level controllers and their locally controlled transport network.

Within one control domain, the transport network topology is auto-discovered by the discovery agent (DA) using the mechanisms described in [ITU-T G.7714] and [ITU-T G.7714.1]. The DA reports the discovery result to the SDN controller for that domain.

When the control domain contains two inter-connected administrative domains, the links between them are often manually configured in the controller. In some scenarios, auto-discovery may be possible e.g., for an Ethernet link, link layer discovery protocol (LLDP) may be enabled to operate across the administrative boundary.

The controller's view of its resources should be kept current since these may change over time due to failure, recovery, network build-out or administrative action.

## 14.1 Creation of network topology by auto discovery procedure

The auto discovery of links and neighbours happens in the transport network. Any in-band communications used for the auto discovery procedure defined in [ITU-T G.7714] and depicted in [ITU-T G.7701] happen across the layer adjacency between subnetworks under the control of the discovery agent (DA). The results (i.e., the observed adjacency to a specific far link endpoint, identified using transport forwarding end port (as defined in [ITU-T G.800]) identifiers) are reported to other components in the SDN controller (DA to TAP, then TAP to LRM which located in SDN controller).

As shown in Figure 14-1 DA1 is responsible for the link end on subnetwork 1 and DA2 is responsible for the link end subnetwork 2. Through discovery messaging, DA1 and DA2 may discover the adjacency between the link ends. This and the capabilities of link are reported to the link end's TAP. The TAP maps the identifiers from the transport name space into the control name space used by the LRM and RC. This information contributes to the network topology graph created in the SDN controller.

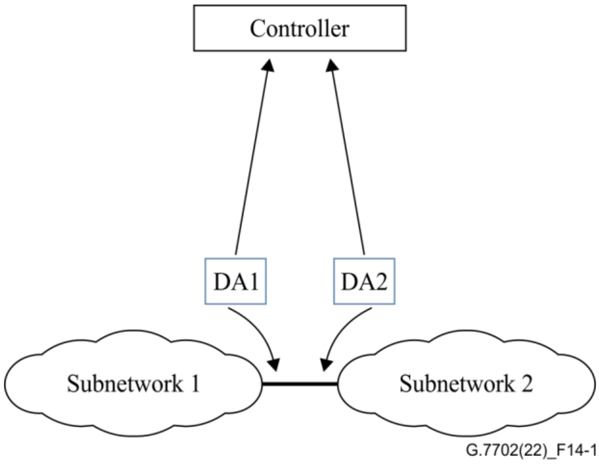


**Figure 14-1 – Example of auto discovery between subnetworks**

When multiple controllers are in use, the DAs need to determine which controller to notify about the link and its capability. Since a controller provides connection management across the link and needs to configure both ends of the link, the appropriate controller is the lowest level controller that has visibility to both ends of the link. If a DA does not have a direct association with the appropriate controller, the link discovery notification may be proxied by a lower level controller.

## 14.2 Creation of abstracted network topology

An abstracted network consists of subnetworks and links, which are created based on assignment of resources from the network topology known to the SDN controller, with consideration of policy, SLA, security, etc. The non-abstracted and abstracted subnetworks as well as links are equivalent with each other, aggregated or sliced from actual transport network resources or abstracted resources, as shown in Figure 14-2 as an example of discovery and creation of abstracted network.

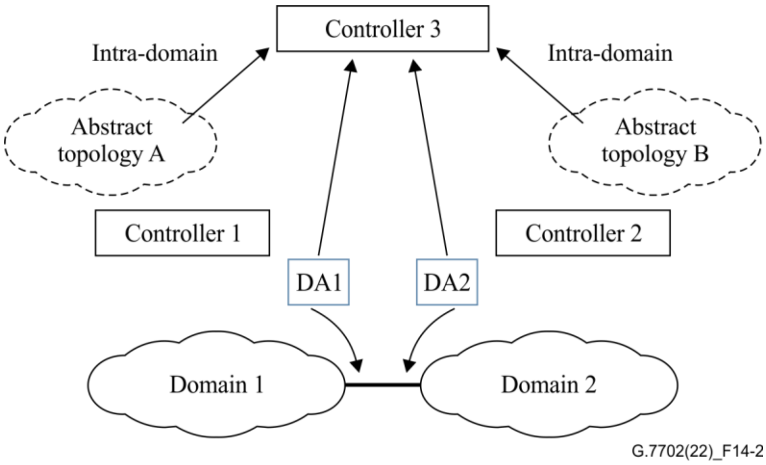


**Figure 14-2 – Example of discovery and creation of abstracted network**

In Figure 14-2 Controller 3 creates an abstracted network topology from network topologies 1 and 2. These topologies are auto discovered separately following the procedure described in clause 14.1. The DAs for network 1 notify Controller 1 of discovered links so it may assemble the topology of network 1 and the DAs in network 2 notify Controller 2 of the discovered links so it may assemble the topology of network 2. Based on the topology for network 1, Controller 1 creates the abstracted network A and reports it to Controller 3 by the CPI interface between Controller 1 and 3.

The physical link between domain 1 and 2 can be configured manually or auto discovered by cooperation of the DAs for Domain 1 with Controller 1 as well as the DAs for Domain 2 with Controller 2. Thus controller 3's total network topology consists of abstract topology A, abstract topology B and the links between these two topologies.

## 15 Controller interactions

There are many possible combinations of controller interactions enabled by the architecture, particularly when considering the dimensions of single/multi resource layers and single/multi-level resource views. This clause describes component interactions for a controller that can configure the FP name space for resources. Other dimensions of recursion are then described including when the controller has only SNP/SNPP names spaces for its view of resources, when multiple server controllers are needed for a concatenated connection and when an interlayer call is needed. Once the recursive cases are described, combinations of them are understood to cover many possible arrangements.

The semantics of multi-layer and multi-level are now described, followed by the types of controller interactions driven by the dimensions of recursion.

The basic types of controller interactions are:

1) Request to a controller where called MC components have FP name spaces in scope. The CC, TAP and DA components have interfaces to the resources and can configure forwarding.

2) Request to a controller where called MC components do not have FP name space in scope. Any connections computed with the SNP/SNPP name space require further resolution to FP name spaces. This triggers a recursive call to a server controller that supports the VN representation.

3) Request that triggers sequential (horizontal) calls culminating in a concatenation of subnetwork connections. Multiple server controllers are called to return subnetwork connections that are assembled in the calling controller.

4) Interlayer request for connection that becomes a link in the VN of the client controller. The [ITU-T G.7701] NCC's "Client NCC coordination in" interface returns SNPs in the client context so adaptation is performed in the server controller.

5) Interlayer mapped server. A server controller with server layer resources sets up a connection that is presented as a pair of SNPs in the client controller.

### 15.1 Interaction type 1

Since the controller involved in this type of interaction has the resource name space in scope and can configure forwarding no recursive calls follow from call/connection setup.

#### 15.1.1 Call and connection control

The interaction of MC components within single layer network for call/connection control including NCC, DS, CC, RC, DA and LRM in a single level controller is illustrated in Figure 15-1. Here the TAP, DA and CC access the FP name space of the transport resources and can configure forwarding.

**Call/Connection set-up process**: The calling/called party call controller (CCC) or NCC-1/NCC-2/NCC-3 with the client context sends the call request to the NCC with server context in controller n (# 1a, 1b and 1c). The NCC sends a client/server context mapping request to the Directory Services (DS) component (# 2) for identifier translation and mapping of virtual network name spaces. The NCC then transforms the call request (with all required information including SLA related performance) to a connection request to the CC (# 3) based on the Identifiers in the server context. The CC sends a route query to the RC (# 4) for a path. The RC computes a path to CC based on the topology and local connection status information which was provided by the LRM (# 5, # 6). Then CC sends the link connection request to LRM for transport network resource allocation (# 6). The LRM interacts with TAP (# 7) to configure the adaptation and termination functions that terminate a link. TAP has interfaces to resources (# 8) that use the FP name space, and these are not exposed SDN interfaces. This enables forwarding in the network resources.

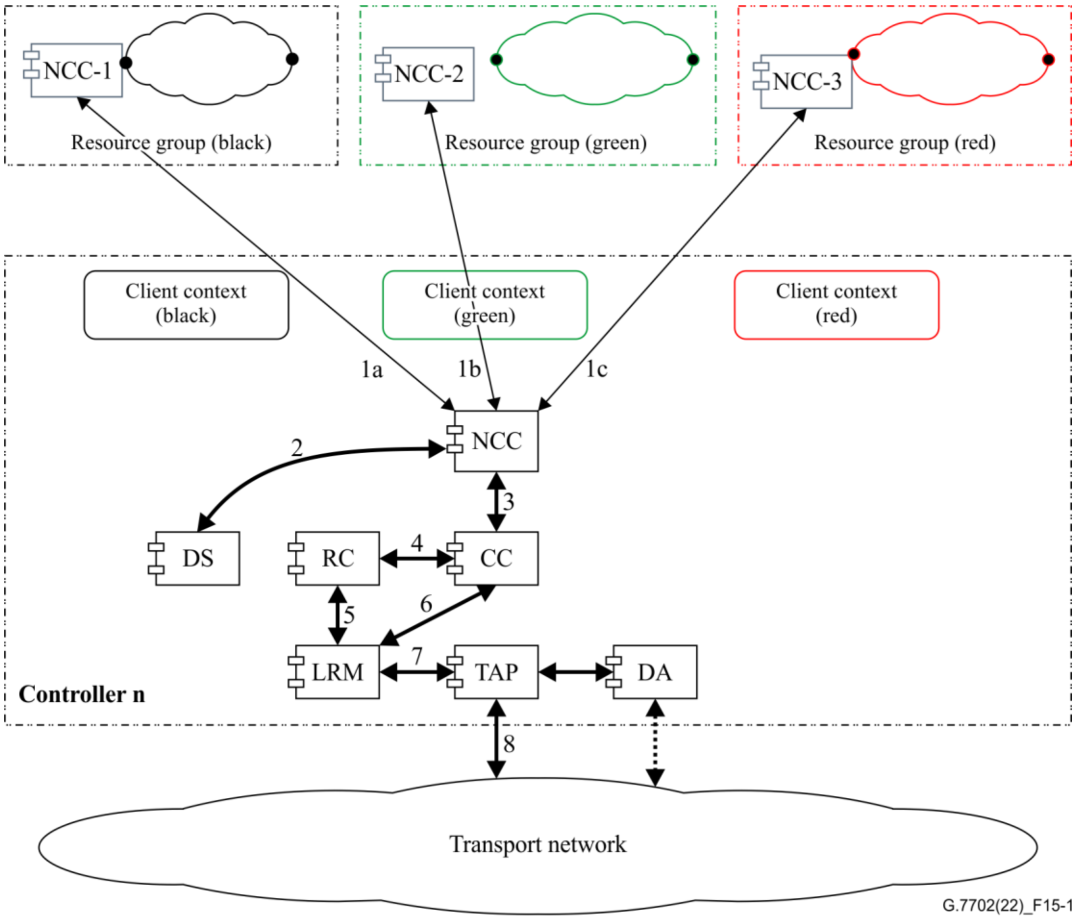


**Figure 15-1 – Call and connection control within single level control for single layer network**

**Connection status maintenance process**: When the connection status is affected by a defect in the transport resources that are in the scope of the controller, the LRM reports the local connection status to the CC and then to the NCC through the network connection status reporting interface. The NCC in the server controller queries the DS to obtain the mapping of the connection identifiers into the client contexts. The NCC in the server controller is responsible for sending the connection status update information to the NCCs in client controllers. If high level SDN controllers or external applications have subscribed to receive connection status information (#9), then this is provided by the notification component.

### 15.1.2 Resource discovery, virtualization (name spaces and mappings) and resource view maintenance

The interaction of MC components within single layer network for resource virtualization and resource view maintenance including NCC, DS, RC, LRM, TAP and DA in a single level controller is illustrated as in Figure 15-1.

In Figure 15-1, the DA component is shown inside the boundary of an SDN controller. This indicates that the DA is one of MC components supporting the SDN controller. DA deployment is not required when physical network resources are manually configured. Therefore, the DA is an optional MC component of an SDN controller. Also, a DA may be deployed in a non-SDN environment. The DA is described in clause 8.3.5 of [ITU-T G.7701].

**Resource discovery process**: The LRM, TAP and DA interacts together to perform the resources discovery function. DAs at the end of a link discover the trail FP to FP resource information over that link through in-band communication. The relationship between two client FPs (cFPs) over the two server FPs (sFPs) can be inferred as defined in [ITU-T G.7714.1]. In general, multiple client FPs (cFPs) in different layers may be associated with one server FP (sFP) using the flexible adaptation function. The TAP is responsible for maintaining the relationship between the cFP and sFP, including the mapping of the namespace between SNPs and FPs. LRMs interact with TAPs and maintain the corresponding SNP to SNP relationship for links in the SNP name space. The LRM reports the local topology (nodes and links), resources and abilities information of transport network to the RC and constructs the underlying network topology and resource database.

**Virtualization within a single level**: Virtualization can be performed within a single level based on the underlying network topology and resources database, which is maintained by the controller, to create/modify/delete VNs for a client. A controller has one RC instance for each client and server context. The RC in a server context maintains the resource information obtained from the LRMs, while the RC in a client context maintains the VN topology information for that client. The virtualization function in the controller maintains separate RC instances to construct the topology required by each client context. During this virtualization process, the RC may interact directly with the LRM to establish the availability of lower level resources and, if necessary, provide partial lower level resources to its client based on policy; or the RC may interact with CC and NCC to first establish a connection and then use this connection as the input to the RC in client context, then use it as a link. A separate name space mapping between the client context and the server context is required for each RC instance.

**Resource view maintenance process:**

In the initial state, a common pool of transport resources under the control of an administration are abstracted into virtual resources and then allocated to different clients, each with their own client context, for their dedicated usage. When a client context is deleted, the virtual network resources dedicated to that client should be deleted and the corresponding SNPs and FPs in the transport resources should be released.

Once the topology and resources state change due to failure, recovery, network build-out or administrative action, the TAP will update the SNP name spaces and also the relationship between the cFP and sFP. The LRM will update the corresponding SNP to SNP relationship for links in the SNP name space and output the changes in local topology (nodes and links), resources and abilities information of transport network to the RC in the server context. With the support of DS to perform the virtualization function per client, the corresponding RC instance will input all the changed information to the RDB and notify the VN client of the virtual resource changes.

## 15.2 Interaction type 2

Consider the controller n+1 in Figure 15-2 and a request to set up a connection within a local transport network 11. Controller n+1 views transport network 11 as a VN with an SNP/SNPP name space. It can compute a route in this name space but as its MC components do not have visibility of the FP name space it thus must request controller n′ to make a connection. Controller n+1 would utilize its NCC, CC and RC, but not LRM and TAP as there is no visibility of the FP name space. The NCC in controller n′ would return a pair of SNPs to the NCC in Controller n+1.

This style of interaction drives recursion up/down a hierarchy of controllers. It covers the case where a client has a VN projected from a subset of resources and/or that is an abstraction of resources. After the controller processes the request, further requests to other controllers may be initiated (types 1,3,4 and 5).

## 15.3 Interaction type 3

This interaction type drives horizontal recursion to perform concatenation of calls/connections between controllers. This style of interaction drives recursion across controllers. It covers the case where a client has a VN consisting of resources projected from multiple server controllers.

Consider the controller n+1 in Figure 15-2 and a request to set up a connection from transport network 11 to transport network 12.

### 15.3.1 Call and connection control

A two level control hierarchy is given in Figure 15-2 as an example to illustrate an end to end call and connection control across two subnetworks and their interconnected links. The end to end connection is divided into three parts: the first part is within the scope of controller n′; the second is within scope of controller n′′; the third is the inter-connection between network 11 and network 12, which is within the control scope of controller n+1.The interaction of MC components between the two levels and within each level are both illustrated and described below.

**Connection set-up**: After the discovery and creation of abstracted network resource process for multiple domains in high level controller as described in clause 7.2.2, Controller n+1 found that the end to end connection set-up and release process need to be divided into three subnetwork connections, Controller n′ is responsible for the connection set-up and release process within network 11, Controller n′′ is responsible for the connection set-up and release process within network 12 and Controller n+1 is responsible for the inter-connection between network 11 and network 12. NCC in Controller n+1 separately sends the call request to the NCC-11 in Controller n' and the NCC-12 in Controller n′′. Interworking between the two levels of controllers is performed via NCC to NCC interaction. Inside each controller, the internal procedure for call and connection configuration is performed as the same component interaction sequence (including NCC, DS, CC, RC and LRM) as described in clause 10.1.1.

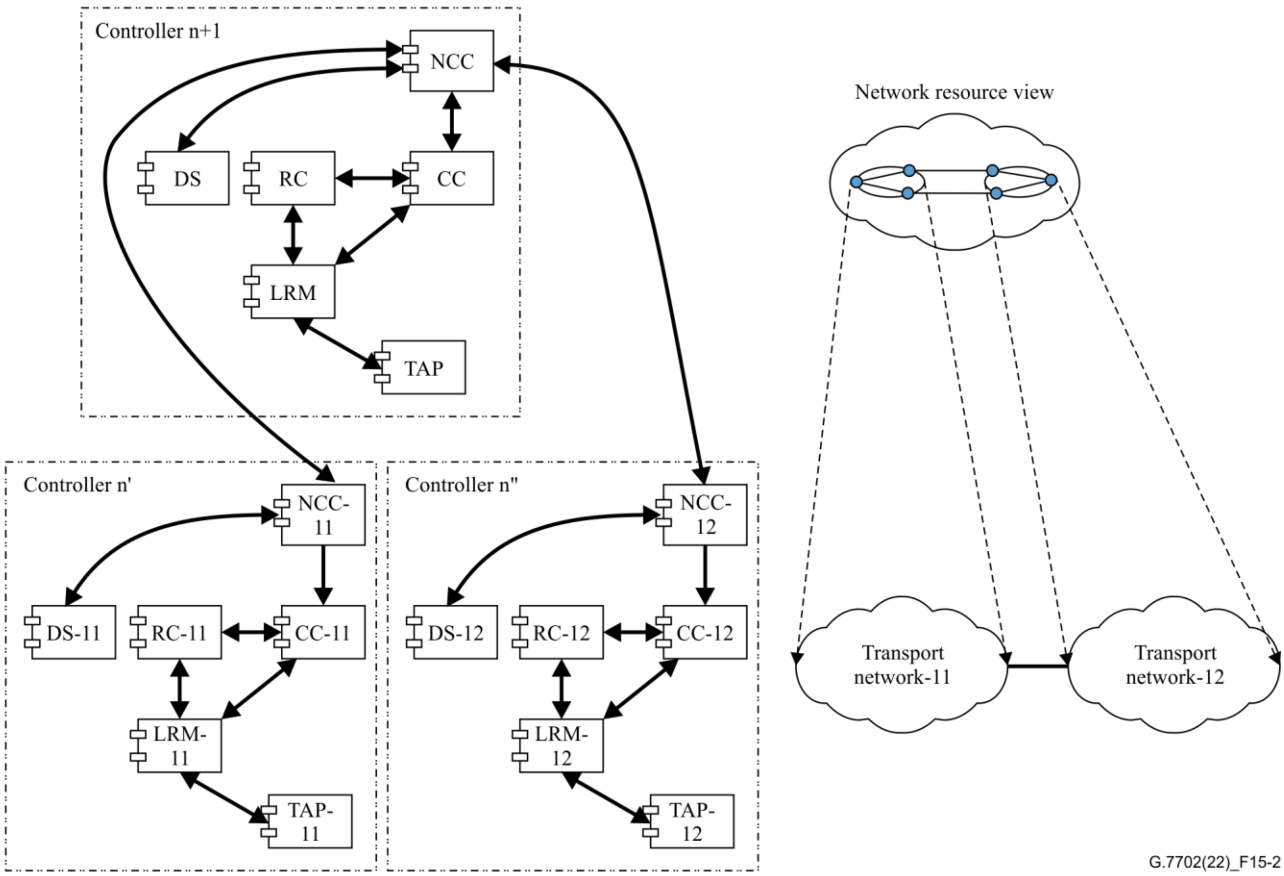


**Figure 15-2 – Connection set-up and release process for two level controllers**

**Connection status maintenance process**: The NCC-NCC instance is also involved in the connection status maintenance process. The LRM reports the local connection status to the CC and then to the

NCC through the network connection status reporting interface. The NCC in the server controller is responsible for sending the connection status information to the NCCs in client controllers.

## 15.4 Interaction type 4

Consider the controller 2 in Figure 15-3. It makes a request for a connection from its server controller 1. The RC in Controller 2 is aware of the adaptation potential of the link to the resource view of subnetwork C in Layer Y. The returned connection is in layer X and controller 2 views it as a link in layer X that becomes part of subnetwork B. The connection is able to use the new link in layer X. The Client NCC Coordination In interface of the [ITU-T G.7701] NCC is used. When the NCC in controller 1 receives the request, it proceeds with the same steps 2-8 as depicted in Figure 15-1. The reply to the NCC in Controller 2 over the Client NCC Coordination In interface provides a pair of SNPs in the calling layer (i.e., layer X). As described in [ITU-T G.7701], the configuration of adaptation between [ITU-T G.800] layers is performed in the server NCC which is the NCC in controller 1.

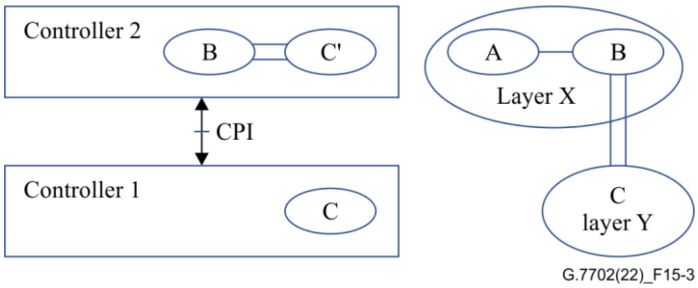


**Figure 15-3 – Client NCC Coordination In interface**

When controller 1 is processing the request, further recursive calls may be needed. For example, interaction types 2, 3 and 4.

## 15.5 Interaction type 5

Consider the controller 2 in Figure 15-3. For a mapped server call, Controller 2 needs to have a multi-layer topology view. This can be accomplished by having a VN representation of subnetwork C which is from a different layer than subnetwork B. The multilayer topology is use by an RC in Controller 2, to calculate a route through both layer networks. The NCC-NCC call is made from Controller 2 to controller 1 using boundary resource identifiers in the context of subnetwork C.

The returned SNPs are in layer X and controller 2 views it as the location of an adaptation into Layer Y. The connection in Layer X is able to use the connection that is created in layer Y. This is the mapped server case described in clause 7.8 of [ITU-T G.7701].

The same steps as described in clause 15.4 are performed when the NCC in Controller 1 receives the request however the SNPs returned are not added as a link to the topology in Controller 2.

When controller 1 completes the request, further recursive NCC-NCC calls of interaction types 1 through 5 may be needed.

# 16 Scalability considerations

## 16.1 Scalability of the controller

The controller can be scaled vertically and/or horizontally.

Vertical scalability can be achieved using a hierarchical stack of controllers; using this recursive application of SDN controllers, the SDN control layer can be easily scaled to cover large networks.

Recursion is illustrated in Figure 11-2. The number of layers employed depends on the size of the network. Increasing recursive depth is used for larger networks.

Horizontal scalability can be achieved using multiple parallel SDN controllers (see SDN controller 1, SDN controller 2 and SDN controller n as shown in Figure 11-4).

Furthermore, the scalability of a SDN controller can be extended by the implementation of controller stacking as long as the network performance is acceptable.

### 16.2 Scalability of components

The scalability of a controller could be enhanced by implementing multiple instances of the same type of MC function and load balancing between those instances inside that controller. This approach is an implementation option of the architecture, other implementations are not excluded.

# Annex A

## Use of the CIM to represent resources

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

Information models provide management views of the resources that need to be controlled and managed. The relationship between both kinds of resources and instances of resources is specified by the information model. The transport network resource data stored in the databases of SDN controllers may be structured into data models (that are generated from information models). In a model-driven architecture CPIs are described by data models. For each client or server context, there is, conceptually, a (logical) database in which the data is structured into data models that are generated from the information model. These data models describe the information that is encoded in a protocol and passed over the CPI as shown in Figure A.1. It may be necessary to map between the representation in the SDN controller resource database and the information in the database of a client or server. For example, it may be necessary to map between the name space used by the SDN controller and the name space used in a server context or client context. Further, it may be necessary to map between the artifacts used in the SDN controller and the (possibly more abstract) artifacts used in the client context as described in clause 7.4.

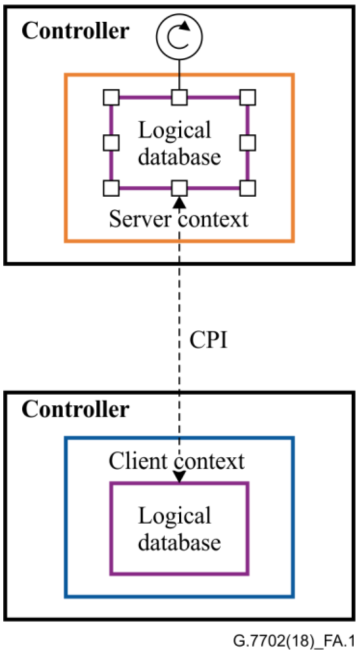


**Figure A.1 – Data model generated from CIM passed over CPI**

Further, if the network resource database in the SDN controller is constructed from the resource database in a number of server contexts it may be necessary to translate between the semantics of these databases. This semantic translation can be avoided if these resource databases are generated from a common information model e.g., ITU-T G.7711.

For the case where network resources are managed by a SDN controller and legacy management systems (EMS/NMS) simultaneously, it is necessary to synchronize the network resource databases in the SDN controllers and EMS/NMS. Data synchronization can be implemented by data query or database synchronization. Data query is implemented by calling the data query interfaces inside the CPIs provided by MCS. Database synchronization is implemented by capturing the data changes of network resource databases and synchronising the changes between the databases of management/control systems according to the mapping relationship between the different information models of the network resources. Both methods require the mapping relationship between the different information models of the network resources.

It should be noted that to fully map and interpret between two IMs, one IM has to be a super set of the other one.

# Bibliography

[b-ITU-T G.872] Recommendation ITU-T G.872 (2019), *Architecture of the optical transport network*.

[b-ITU-T G.7715] Recommendation ITU-T G.7715/Y.1706 (2002), *Architecture and requirements for routing in the automatically switched optical networks*.

[b-ITU-T M.3400] Recommendation ITU-T M.3400 (2000), *TMN management functions*.

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

## SERIES OF ITU-T RECOMMENDATIONS

| | |
|---|---|
| Series A | Organization of the work of ITU-T |
| Series D | Tariff and accounting principles and international telecommunication/ICT economic and policy issues |
| Series E | Overall network operation, telephone service, service operation and human factors |
| Series F | Non-telephone telecommunication services |
| **Series G** | **Transmission systems and media, digital systems and networks** |
| Series H | Audiovisual and multimedia systems |
| Series I | Integrated services digital network |
| Series J | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K | Protection against interference |
| Series L | Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant |
| Series M | Telecommunication management, including TMN and network maintenance |
| Series N | Maintenance: international sound programme and television transmission circuits |
| Series O | Specifications of measuring equipment |
| Series P | Telephone transmission quality, telephone installations, local line networks |
| Series Q | Switching and signalling, and associated measurements and tests |
| Series R | Telegraph transmission |
| Series S | Telegraph services terminal equipment |
| Series T | Terminals for telematic services |
| Series U | Telegraph switching |
| Series V | Data communication over the telephone network |
| Series X | Data networks, open system communications and security |
| Series Y | Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities |
| Series Z | Languages and general software aspects for telecommunication systems |

Printed in Switzerland
Geneva, 2022