OPEN NETWORKING
FOUNDATION

SDN Architecture for Transport Networks

March, 15, 2016

ONF TR-522

OpenFlow™

ONF Document Type: Technical Recommendation
ONF Document Name: SDN Architecture for Transport Networks

Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any marks and brands contained herein are the property of their respective owners.

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

Table of Contents

List of Figures

1 Scope and Introduction

This document describes the application of SDN architecture [Arch, Arch1.1] and Information Modeling [CIM] to transport networks. Transport networks need to accommodate growing bandwidth demand from data centers, support rapid service deployment by service providers, and provide real-time responsiveness to capacity/QoS changes. Application and service providers want the ability to request and provision edge-to-edge connections with guaranteed SLA (in terms of e.g., bandwidth, delay, availability, error performance) over multiple types of transport infrastructures, including OTN, MPLS-TP and Carrier Ethernet. The SDN architecture and Information Model applied to transport networks supports key features for transport such as abstraction/virtualization and the relationship of flow and connection.

Specific goals of this document are to address:

a) the application of SDN architecture in carrier transport networks
b) the use of the SDN architecture to support virtualization of transport network resources
c) the use of SDN architecture to support integrated control of diverse multi-domain, multi-layer transport network organization
d) characteristics of transport connectivity services relative to a basic unidirectional data flow model and how these relate to the SDN architecture
e) common behaviors of transport network equipment such as monitoring and protection and how these can be accommodated in SDN architecture

2 Abbreviations and Conventions

This document uses the following abbreviations and acronyms:

API	Application Programmer's Interface
CPI	Controller plane interface
MO	Managed object
MPLS-TP	Multi-protocol label switching, transport profile
NBI	Northbound interface
NE	Network element
OAM	Operations, administration, maintenance
ODU	Optical Data Unit
OFS	OpenFlow-Switch protocol
ONF	Open Networking Foundation
OTN	Optical transport network
P2P	Point-to-point
PM	Performance monitoring
RG	Resource Group
SDN	Software-defined networking
SNC	Subnetwork connection
TCM	Tandem connection monitoring
VPN	Virtual Private Network

This document uses the following terms defined in existing standards:

Connection	Refer to [G.805]
Connection port	Refer to [G.805]
Connection point	Refer to [G.805]
Matrix connection	Refer to [G.806]
Subnetwork	Refer to [G.805]
Link	Refer to [G.805]

3 Architecture of SDN for transport networks

3.1 Application of SDN Architecture and Information Modeling

The architecture of SDN is specified in the ONF SDN architecture document [Arch, Arch1.1], which identifies core principles of SDN and applies them to architectural components and interfaces. This document delves further into the SDN control of transport network resources, and exposure of the abstract network resources and states to their respective SDN controller. It does so by further elaborating on the high level representation of abstract network resources in the SDN Architecture document via utilization of transport network and network element architectural models.

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

In order to simplify the description, transport network dataplane modeling in ITU-T G.805 [G.805] utilizes the concepts of layering and partitioning within each layer network, in a manner that allows a high degree of data plane recursion.

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

The components of the transport network architectural model can be characterized as topological components, transport processing functions, and transport entities.

- The topological components provide the most abstract description of a transport network in terms of the relationship between sets of connection points within a layer network.

- Transport processing functions are used to model the processes – implemented within equipment – that manipulate the information that is being transferred across the transport network, as well as OAM information that is inserted, extracted and processed within the transport network.
- Transport entities provide the means to transfer information across the transport network, between connection points. Transport entities are configured within topological components.

The Information Modeling project in ONF has defined a Core Information Model [CIM] to be used to model the management and control of network components in a technology-independent way. The Core Information Model is extended with fragments for specific technologies such as OTN and Ethernet. The resulting combination can then be mapped to a specific control interface.

Transport network components and entities fit cleanly into the Core Information Model. The topological components and transport processing functions of the transport network are modeled in the Core Information Model using the ForwardingDomain and LogicalTerminationPoint object classes. Transport entities in the transport network are modeled using the ForwardingConstruct object class in the Core Information Model.

3.2 Use of Abstraction and Virtualization in Transport Networks

The SDN architecture encompasses the Application Plane, Controller Plane, and Data Plane whereby the SDN controller manages data plane resources, and applications receive services from the SDN controller. Within this document, we are focusing on the control of transport data plane resources. As described in the SDN architecture, the controller plane may involve multiple hierarchically arranged, logically centralized SDN controllers. At the lowest level of recursion, data plane resources are exposed by physical network entities (NEs). At other levels in the recursion, data plane resources are exposed to the client controller by the server controller; i.e. the data plane resources being managed by a controller need not be directly supported by physical NEs (e.g., they may be virtual resources). Thus, control of abstract transport network resources or resource groups [Arch1.1] may encompass both "physical" and virtual resources. In either case a client context provides an abstract view of the resources. The Controller Plane Interface is the generic interface across which an instance of the SDN information model is managed. As the SDN architecture operates on an abstract model of the data plane, there is no architectural distinction between the control of physical and virtual.

For illustrative purposes, we start with a simple example of two controllers, where each controller belongs to a different provider. Figure 3-1 illustrates SDN controller Green's and SDN controller Yellow's control of their respective physical NEs.

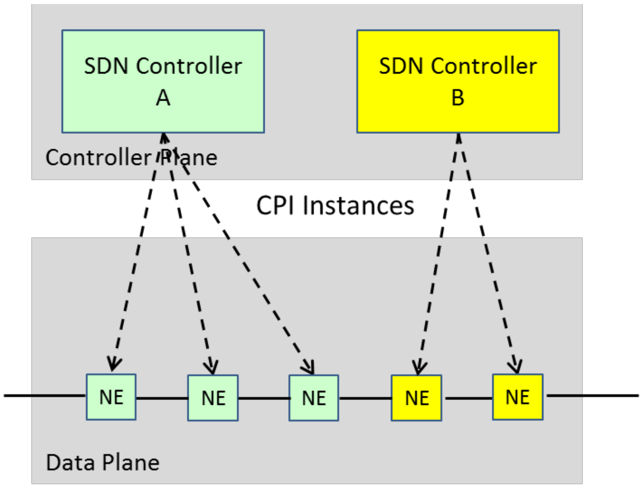

Figure 3-1: Example of two controllers managing their respective physical NEs

This is a very simplified figure, as operations take place on an abstracted view of the physical resources, which has not been illustrated. Specifically, there are client contexts that represent the controlled entities (here the green and yellow physical NEs), and the Green and Yellow controllers act upon these client contexts, respectively.

Now consider a two level controller hierarchy with SDN controller Blue offering virtual network resources to SDN controller Green. Consequently, Green’s data plane now includes its physical NEs as well as virtual resources provided by Controller Blue. From SDN controller Green’s perspective, everything to its south is in its data plane. This is illustrated in Figure 3-2 below, which explicitly illustrates the associated client and server contexts.

> Note – In Figure 3-2, NE1 and NE2 represent physical NEs supporting Resource Groups 1 and 2 for SDN Controller G, while SDN Controller A exposes resources as two independently managed Resource Groups (i.e., “Virtual NEs”) to Controller G. Different client/server associations/sessions are assumed to be needed to access different Resource Groups exposed by Controller A. This might be appropriate, for example, if businesses Green (with Controller G) and Gold (with Controller A) had two completely independent business relationships with each other. SDN Controller B on the other hand exposes a common Resource Group B with a substructure or topology of 5 Resource Groups (RG B1-5) and an associated set of connecting links. The client contexts contain all of the information exposed to a particular client and the controlling entity – here Controller G – sees no distinction between physical and virtual resources.

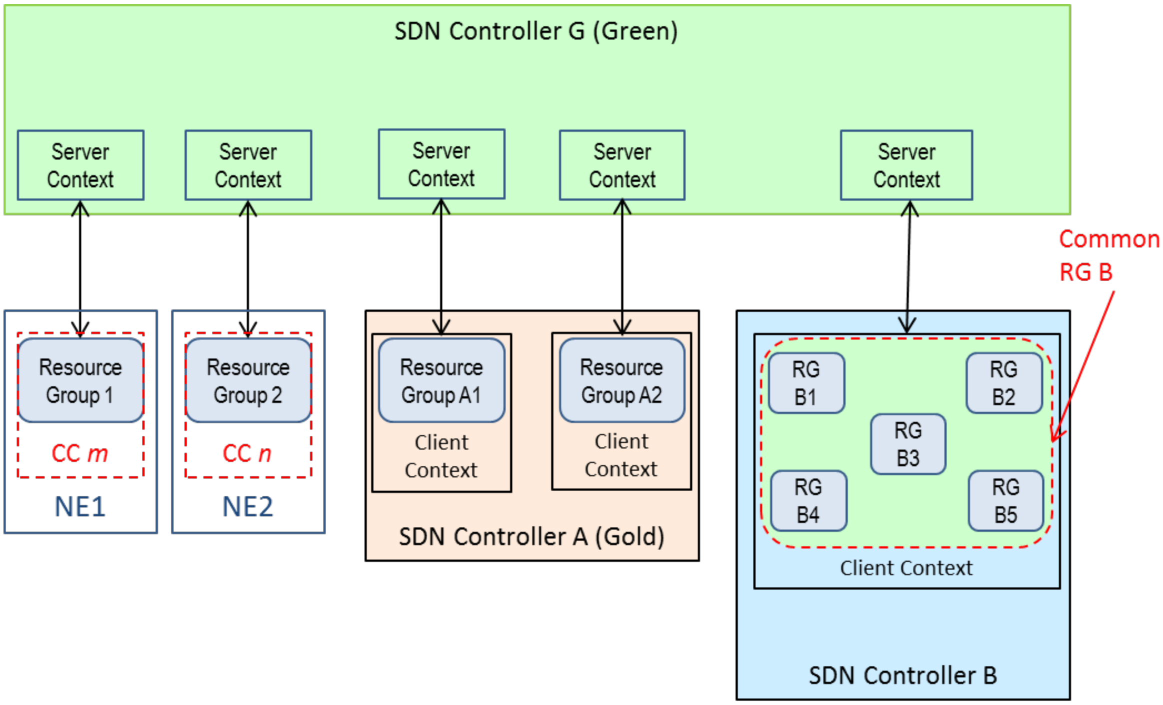

Figure 3-2: Example of control of data plane involving physical and virtual resources

Controller B in Figure 3-2 offers a Transport API to Controller G that supports retrieval of topology information in the Client Context for Controller G and the ability to create and control connectivity services across the common resource group exposed to Controller G. Controller B has the knowledge of the relationship between the virtual network exported to Controller G, the underlying model of resources allocated from B to G, and the mapping to B's resources. This is illustrated in Figure 3-3, below.

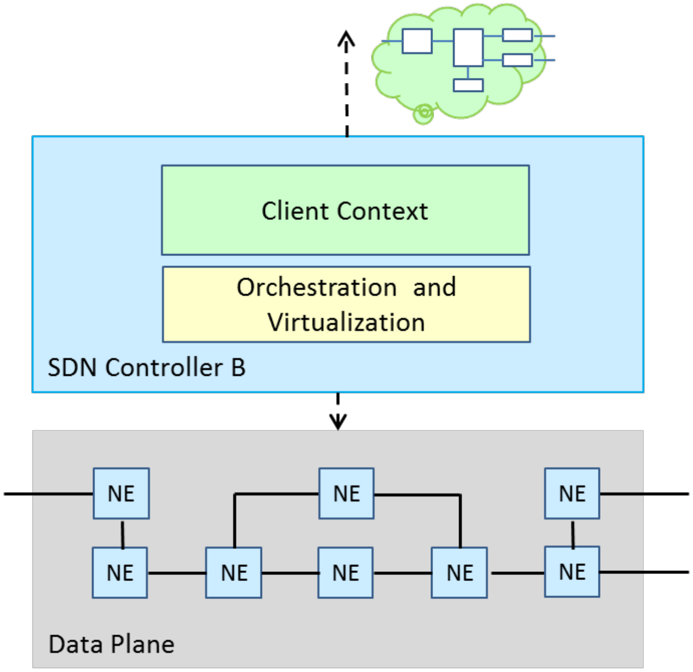

Figure 3-3: Example of relationship of virtual and physical resources

In general, the following points should be noted:

- A 1:1 relationship across an instance of the CPI does not imply that the resources are directly contained within a single NE; e.g., a single Resource Group provided by SDN controller B and visible to G may be supported by multiple NEs and connecting links in B's view of the resources.
- There may be an m:n relationship between the Resource Groups presented to G and the NEs that B manages directly.

3.3 Modeling of Multi-domain and Multilayer in Transport Networks

3.3.1 Multi-domain and Multilayer Networks

Service Provider Transport Networks are commonly partitioned into multiple administrative domains for scaling and operational concerns. The SDN architecture supports a model of integrated control of multiple domains primarily using hierarchical organization of SDN Controllers, including a Parent Controller (sometimes called a Network Orchestrator) and Child Controllers, as shown in Figure 3-4 below.

Multiple administrative domains are also found in multi-provider scenarios, where each provider runs and operates separated resource infrastructures (either physical or virtual). The relationship between controllers in those separated domains is not necessarily hierarchical. Also, the level of information exposure differs in the multi-provider case with regards the single provider environment. Collaboration among domain controllers will be required to enforce services across domains. Schemes for such collaboration are matter of further work. Peer collaboration between Controllers is discussed in [Arch1.1] section 8.3 and will be a future work item.

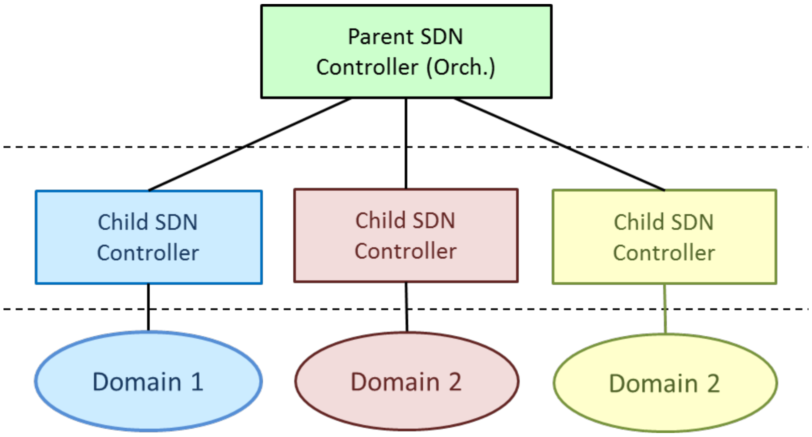

Figure 3-4: Hierarchical Controllers in Multi-domain Network

Multilayer networking can also be implemented with control layer integration of domains that are administratively separated and operate internally at different switching layers, for example integration of IP and Optical domains under a common control layer.

This integration of multilayer networking can be modeled using the hierarchical control of multiple administrative domains as in the SDN Architecture document [Arch] section 4.3.6, where multiple Child Controllers are coordinated through a common Parent Controller as shown in Figure 3-5 below.

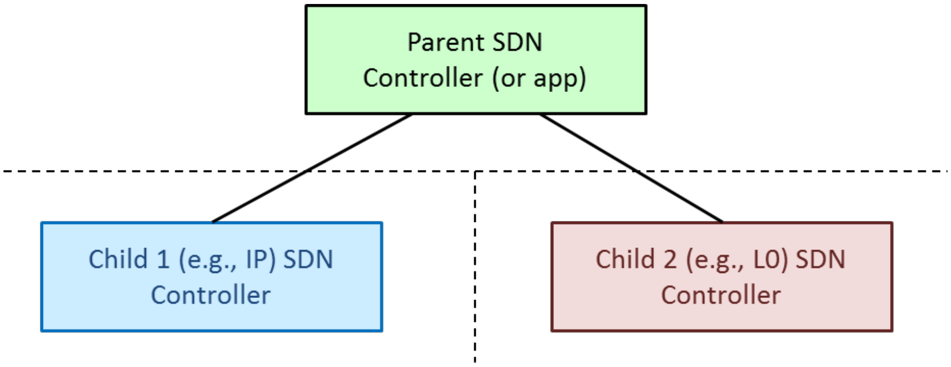
Parent SDN Controller (or app)
Child 1 (e.g., IP) SDN Controller
Child 2 (e.g., L0) SDN Controller

Figure 3-5: Layer Domains with Separate Controllers

The Parent Controller has access to topology and status information for each server domain and can coordinate actions across both domains. Multilayer path computation and optimization can be provided as a function of the Parent Controller or as a separate function.

3.3.2 Multilayer within a Single Domain

Multilayer adaptation also occurs within single administrative domains. When two domains use different layers of switching technology internally, one domain or the other (or both) must support adaptation from its internal switching layer into another layer supported by the other domain at its ingress and egress interfaces.

Support of this internal adaptation is another scenario of multilayer networking, that is, control of multiple switching layers within a single domain. Two possible use cases are:

- The transport network domain supports an access interface technology that differs from its internal switching layer – an example is support of Ethernet interfaces and Ethernet services that are carried over OTN internally.
- The transport network domain supports adaptation between multiple internal signals within the same technology – an example is the support of internal OTN adaptation for grooming and efficiency purposes, i.e., ODUj-over-ODUk.

In the example shown in Figure 3.6, adaptation from the access interface to the internal layer (yellow to blue in the figure) is shown.

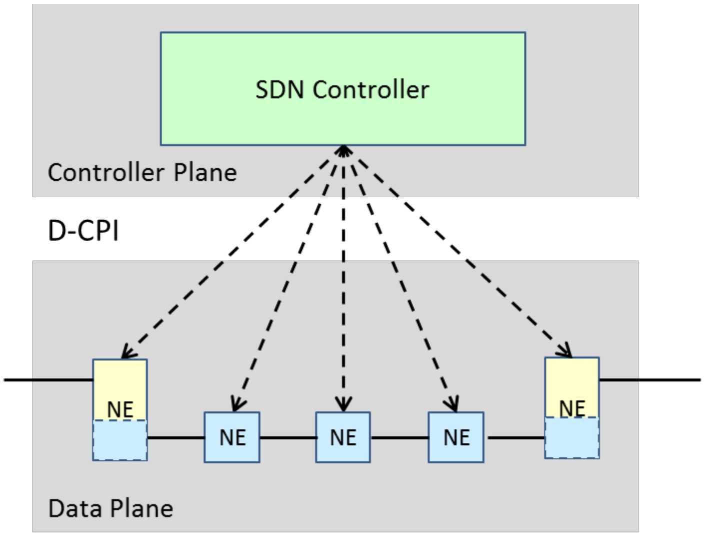

Figure 3-6: Control of Adaptation within a Domain

Multilayer adaptation is modeled in ITU-T Recommendation G.805 [G.805] as a function existing between the client layer and the server layer trail, which supports the transfer of monitored adapted characteristic information of the client layer. Adaptation allows the client layer characteristic information to be put into a form suitable for transport over the server layer trail. An example might be adaptation of packet information into a bitstream suitable for transport over a digital circuit, including delimitation of packet beginning and end and insertion of filler as needed in between packets.

Support of adaptation between layers of a transport network requires the ability to identify the client layer information and control of the specific layer adaptation method to be used.

4 Modeling of Transport Connectivity Services

4.1 Flow vs. Connection

Transport networks are generally designed to deal with “connections”, which are entities that encompass multiple related “flows”. Transport network terminology typically uses the term “connection”, or “subnetwork connection”, as defined in G.805 [G.805]:

- Subnetwork connection (SNC): A "transport entity" that transfers information across a subnetwork, it is formed by the association of "ports" on the boundary of the subnetwork

Note: as discussed in a previous section, a “transport entity” is modeled as a ForwardingConstruct object in the Core Information Model [CIM].

In this document, a “flow” is defined as a stream of data that is identifiable by some criteria such as common packet header values, relative time position or frequency and is acted on and/or forwarded based on a common set of rules within the bounds of a connection. In OpenFlow, for example, a flow is identified by match field values configured into a flow table entry, together with an associated set of instructions and actions.

Flows in this definition are unidirectional in nature. Flows can exist at any layer using match fields associated with that layer, for example layer 1 flows can be identified by specific time/tributary slots and layer 0 flows by specific center frequency and slot size.

Flows can be nested in which case a set of smaller flows can be distinguished within a larger flow. These smaller flows all share the same flow identification criteria of the larger flow and can be distinguished by means of additional flow identification criteria of which the value is unique per smaller flow.

The flow ID may be used to steer the forwarding of the flows within connection X to a specific subset of m out of n output ports of connection X; such forwarding control is referred to as "connectionless" flow forwarding and explicit 'unicast', 'multicast' and 'broadcast' flow forwarding rules are deployed. For the case that the flow ID is not used to steer the forwarding of the associated flow within connection X, the forwarding control is referred to as "connection-oriented" flow forwarding and one implicit 'broadcast' flow forwarding rule is deployed.

> Note that a L2 or L3 connection that supports connectionless flow forwarding is often referred to as a L2 or L3 VPN. From the flow forwarding perspective such connection can be perceived as a virtual (flow forwarding) network.

The OpenFlow Switch (OFS) protocol may be used to control the forwarding of flows within a connection that supports connectionless flow forwarding. For such case, a "flow ID" represents a common value in a subset of the match fields configured into a flow table entry.

4.1.1 Connections and OAM

In the transport network a connection represents a transport entity within which zero or more flows are transported (a common example is a bidirectional connection with a separate flow in each direction). More complex cases include point-to-multipoint, multipoint-to-multipoint and protected connections.

The status and performance of an individual connection is monitored, using OAM flows that are associated with the data flows but are processed within the NEs to monitor status and pass internal status indications and information. The status and performance of the forwarding of an individual flow within a connection is not monitored, however the forwarding of an individual flow may be modified based on OAM information, for example the flow may be terminated and an open connection, locked or alarm indication maintenance signal transmitted instead. The following sections provide more detail on layering of connections, connection life cycle, different connection types and connection protection. At a high level, these considerations require that the SDN architecture allow for local, autonomous processing at the NE in order to handle functions that involve rapid reconfiguration of related flows or monitoring or generation of OAM signals.

4.1.2 Layered Connections

In a layered transport data plane model, when a connection X is supported by a server layer connection Y, the flows within connection X are transported (as an aggregate or individually) within connection Y, but are not individually monitored. The endpoint and client adaptation functions within connection X generate and terminate the flows. Within connection Y, all flows associated with connection X are uniquely identified by an explicit label or an implicit label (i.e. time or frequency) which is inserted and removed in the layer Y to layer X adaptation function. This label acts as a “flow ID” for the set of flows within connection X within the bounds of connection Y and as a “connection ID” for connection X at the edges of connection Y. There is as such a flow ID ⇔ connection ID duality, representing an internal versus external view.

> Note that a connection can have zero active flows; for such case the connection will still exist. A flow can have zero traffic units in flight; for such case the flow forwarding rules will still exist.

4.1.3 OAM Flows

Connections may contain OAM flows in addition to the flows from their client layer connections. The set of OAM flows within a connection is distinguishable from client layer connection flows by means of an explicit label or an implicit label (i.e. time, frequency). In addition, each specific OAM flow within the set of OAM flows is uniquely identified by its own explicit label or implicit label (i.e. time, frequency).

The set of OAM flows within a connection is transferred either in a time or frequency based OAM channel that complements the channel in which client layer flows are transferred, or in the same channel in which client layer flows are transferred. For the latter case, OAM flows and client layer flows (may) interfere with each other. For the former case, the OAM channel may be further divided into smaller (OAM type-specific) channels, each with dedicated channel bandwidth. The OAM channel or a smaller OAM type-specific channel may transfer one or more OAM flows.

Adaptation and trail termination atomic functions within a connection have OAM flow awareness for a subset of the OAM flows. These atomic functions may generate and insert one or more OAM flows and terminate one or more OAM flows. Adaptation functions may also provide higher layer management and control functions access to a subset of the OAM channel bandwidth so that these management and control functions can communicate with each other.

NOTE – Trail termination functions include the OTN, MPLS-TP and Ethernet Physical layer trail termination functions specified in ITU-T Recommendations G.798 [G.798], G.8121 [G.8121] and G.8021 [G.8021] and Ethernet MAC layer Flow Termination (ETH*x*_FT) functions specified in ITU-T Recommendation G.8021 [G.8021].

Interruption of a connection Y interrupts the forwarding of the flows within that connection Y and its client layer connections X, W, etc.

4.2 Transport Connection Life Cycle

Connections created in a transport network typically follow a lifecycle that is designed to validate correct connectivity and ensure that the connection meets service level agreements made with the customer. Many functions are accomplished beyond the basic establishment of flow forwarding such as:

- Establishment and deletion of MOs in the NEs managed object database

- Administrative locking and unlocking
- PM (Performance Monitoring) set up to monitor the end-to-end connection
- TCM (Tandem Connection Monitoring) points set up to monitor performance across segments of the connection

The following actions and notifications typically take place to support the setup and operation of a connection:

Creation and deletion, as well as associated confirmations:

- connection setup/tear down
- creation/removal of monitors and PM counters

Attribute modification (Controller) and confirmation:

- Administrative state modification (locked/unlocked)
- TCM activation/deactivation
- Alarm report control (activate/deactivate)
- PM activation, deactivation, retrieval, reset registers, reset timers
- Delay measurement activation, deactivation, retrieval, reset registers, reset timers
- Protection commands

Attribute change notification (Switch):

- Operational state change notification
- Protection status change

Other functions (which are multi-step operations)

- Add/remove TCM on an existing connection (create/delete second class objects)
- Add/remove delay measurement on an existing connection (create/delete second class objects)
- Activate/deactivate testing

Generic notification service:

- Notification of alarms or fault cause (defects after fault correlation) or “raw” defects. Refer to [G.7710] and [G.806] for a definition of alarm, fault cause and defect.
  NOTE – Transport network elements are required not to support the notification of fault causes and defects in order to prevent alarm storms overloading the data communication network and management systems.

Creation and deletion of the flows of a connection are actions typically understood to be initiated by the Controller via the Data-Control Plane Interface (D-CPI) defined in the SDN Architecture [Arch]. However, to the extent that generation and processing of monitoring overheads, protection switching and generation of notifications such as alarms take place locally on the Network Element, the SDN D-CPI allows for such local processing to be instantiated, enabled, disabled and deleted by the Controller.

4.3 Flow Forwarding Configuration for Different Connection Types

Connections of different types can be described in terms of the forwarding of unidirectional flows from input to output of the switch. The number of flows and their association (e.g., bidirectionally symmetric) depends on the type of connection.

Examples of common flow forwarding behaviors in different connection contexts include.

- Unconnected port
- Point-to-point Connection
- Point-to-multipoint connection
- Multipoint-to-point connection
- Multipoint-to-multipoint connection
- Rooted multipoint connection

Forwarding of flows associated with a particular connection can be affected by OAM or signal monitoring and can involve specially generated signals such as an Open Connection or Replacement Signal being inserted into a flow. Changes to the flow configuration within a connection as well as sending of specially generated signals can be done under the direction of the SDN Controller but with some penalties due to the latency of control communication with the Controller and the overhead of control messaging between switch and Controller.

Because of these disadvantages, signal monitoring, control of associated flow configuration changes as well as generation of special signals are allowed as functions under local autonomous control.

4.4 Flow Behavior in Connection Protection

Flow forwarding behavior can also be driven by connection protection in case of failure. In SDN it is assumed that a protection group is set up, modified and deleted under control of a SDN controller, either during or after connection setup. In the event of a failure, though, flow forwarding must be modified rapidly to keep information flowing along the protection path.

There are many different protection arrangements and behaviors used in Transport Networks. There a number of different protection types and triggers, including protection that may be triggered by an event on a different but related connection.

Linear protection for P2P connections follow the generic architecture defined in ITU-T G.808.1 [G.808.1]. Ring protection for P2P connections follow the generic architecture defined in ITU-T G.808.2 [G.808.2] and technology specific architectures defined in G.873.2 [G.873.2], G.8032 [G.8032] and RFC 7271 [RFC7271].

Mesh protection for P2P connections follow the generic architecture defined in ITU-T G.808.3 [G.808.3].

Within intermediate nodes of a connection, cross-connections on working and transport entities are in general (i.e., assuming no nested protection) configured as in case of unprotected P2P connections as described in the previous section.

At the head and tail ends of the linear protection domain, cross-connections are set up between the connection point associated with the normal traffic and the connection point associated with the protection group, which is used to bridge/select the traffic in idle (i.e., No Request) conditions.

Changes to flow forwarding at the head and tail ends can in theory be done either by a remote SDN Controller or by a local process. However in the remote case the Controller must be first notified by the switch of the failure event, then push down a new flow configuration that compensates for the failure. Due to service requirements on protection times, flow reconfiguration for connection protection will in many cases need to be a local autonomous function using previously downloaded precomputed instructions or local computation in order to avoid the latency of control messaging between the Network Element and Controller.

5 Future Work

This 1.0 version document describes general architecture aspects of transport SDN networks. There will be other aspects of transport SDN architecture that need further study and will be added incrementally in future versions. Possible future work items include automatic neighbor discovery, controller based connection restoration, operation administration maintenance and provisioning (OAM&P), peer controller relationships, multiple administrative domain interactions, wireless-based transport issues, as well as additional details on connection lifecycle, transport connection and flow types and connection protection.

6 References

The following references may be useful background for SDN implementers.

[Arch] ONF TR-502, “SDN architecture, Issue 1”, June 2014, available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

[Arch1.1] ONF TR-521, “SDN Architecture, Issue 1.1”, January 2016, available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf

[CIM] ONF TR-512, “Core Information Model (CoreModel) Version 1.0”, March 30, 2015, available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Core_Information_Model_V1.0.pdf

[G.805] ITU-T Recommendation G.805 (03/2000), Generic functional architecture of transport networks

[G.798] ITU-T Recommendation G.798 (12/2012), Characteristics of optical transport network hierarchy equipment functional blocks

[G.7710] ITT-T Recommendation G.7710 (02/2012), Common equipment management function requirements

[G.8121] ITU-T Recommendation G.8121 (11/2013), Characteristics of MPLS-TP equipment functional blocks

[G.8021] ITU-T Recommendation G.8021 (04/2015), Characteristics of Ethernet transport network equipment functional blocks

[G.808.1] ITU-T Recommendation G.808.1 (05/14), Generic protection switching - Linear trail and subnetwork protection

[G.808.2] ITU-T Recommendation G.808.2 (11/2013), Generic protection switching - Ring protection

[G.806] ITU-T Recommendation G.806 (02/2012), Characteristics of transport equipment - Description methodology and generic functionality

[G.873.2] ITU-T Recommendation G.873.2 (08/2015), ODUk shared ring protection

[G.8032] ITU-T Recommendation G.8032 (08/2015), Ethernet ring protection switching

[RFC7271] IETF RFC 7271, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", June 2014

[G.808.3] ITU-T Recommendation G.808.3 (10/2012), Generic protection switching - Shared mesh protection

LIST OF CONTRIBUTORS

Maarten Vissers
Italo Busi
Malcolm Betts
Lyndon Ong
Guoying Zhang

Many thanks to Eve Varma, Dave Hood and Luis Contreras for their review and comments.