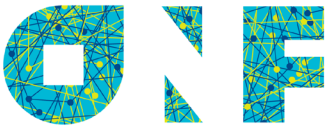


OPEN NETWORKING
FOUNDATION

Optical Transport Use Cases

Optical Transport Working Group

ONF TR-509

ONF Document Type: Technical Paper
ONF Document Name: optical-transport-use-cases

Disclaimer

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

Table of Contents

1 Introduction

A use case describes the actions taken by a hypothetical set of actors as they work to achieve a goal. The actors may be human, but they are often software entities or organizational entities. The purpose of the use case is to present a believable scenario that assists in identifying architectural components, relationships and requirements.

This document applies those principles to the software-defined networking (SDN) control of transport networks. It describes four use cases related to transport networks. Its intention is to guide the development of high-level requirements, architecture, and protocol definition for Transport SDN.

Briefly, the use cases are:

- Direct control of optical components in enterprise networks
- Carrier Ethernet network virtualization
- Service provider data center interconnection
- Packet-optical integration

The use cases in this document are designed to illustrate only some of the possible applications of SDN in the transport network environment. The list is not exhaustive or final; there are many other possible applications and more may be investigated in the future.

The intent is to present and analyze the use cases in a way such that high-level requirements can be derived from them. The use cases are not intended as detailed specifications for realization.

In the interest of brevity, the use cases aim to describe realistic network and operational scenarios without supplying extraneous details; further, the reader should be aware that any details that are presented may not apply to all scenarios.

The document provides sufficient detail to identify differences between existing methods of operation and the use of SDN/OpenFlow™, and identify functional requirements on SDN/OpenFlow to support these use cases.

2 Use Case 1: Photonic Enterprise Network

This use case addresses enterprise data center interconnection by a private all-optical network, where the network is owned and operated by the same enterprise that owns and operates the data centers. It is assumed that the purely photonic part of the network is bounded and supports a limited number of large flows on a wavelength basis. Such photonic islands may be interconnected via OEO equipment for scaling reasons, but the OEO aspect is beyond the scope of this use case. As a result, it is suitable to route on a wavelength basis, assuming that only a WDM network is interconnecting the nodes.

2.1 ENVIRONMENT

Data center services that may be offered by the enterprise to third parties are not included in the scope of this use case, and the network topology is assumed to be known to the SDN controller. As such, no control-virtual network interface (CVNI) is shown in the environment diagram of Figure 1. While only two data centers are shown, the use case allows for any number.

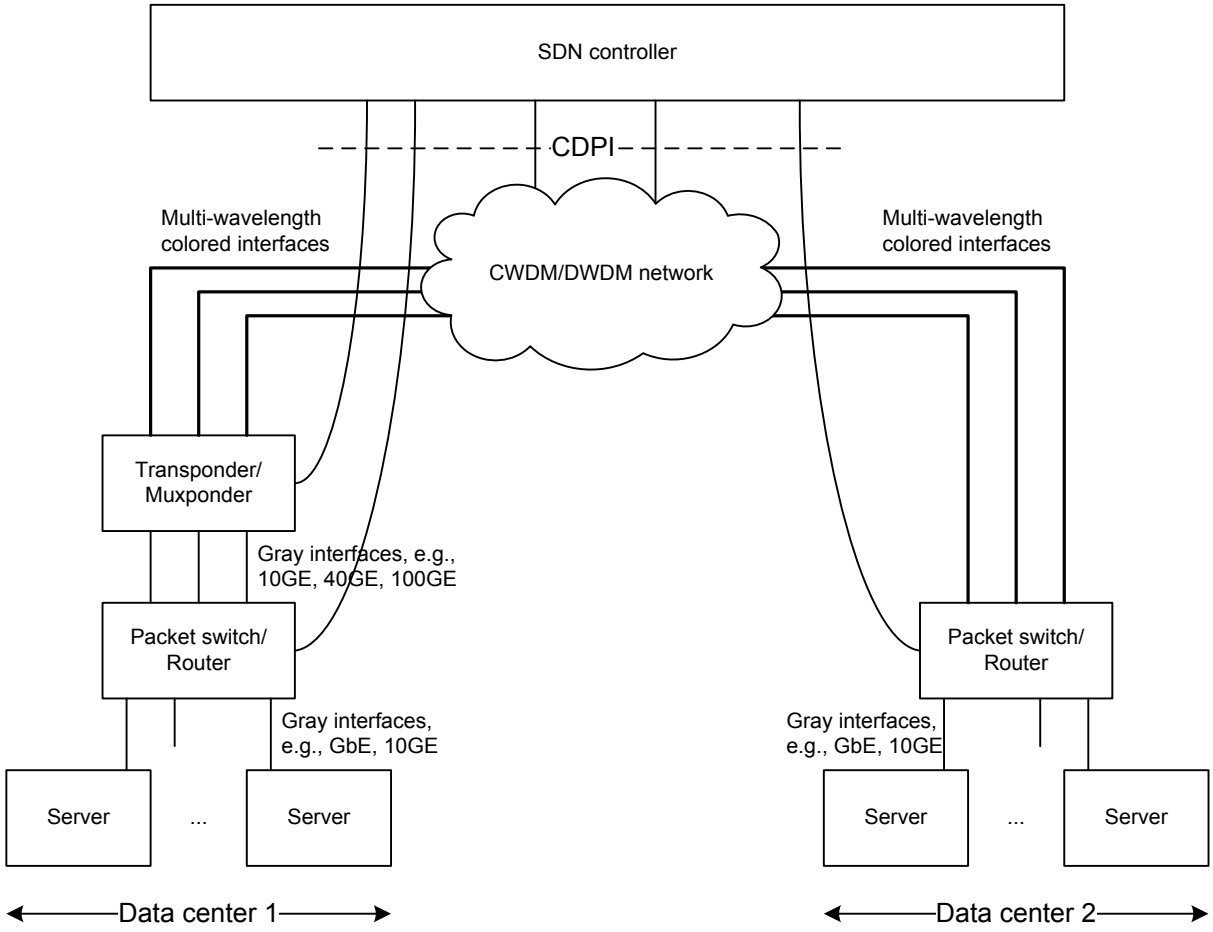


Figure 1 – Data center interconnect, all-optical network

As illustrated in Figure 1, a data center contains a number of servers, where the applications are hosted, and the data center fabrics may consist of routers and packet switches. Two cases are shown. In Data Center 1, the packet switch may have gray interfaces to a transponder or muxponder, which is responsible for providing WDM wavelength signals. In Data Center 2, the WDM wavelengths are provided directly by the packet switch.

In both cases, the fibers enter a CWDM/DWDM network, which comprises an arbitrary arrangement of passive optical devices, active optical devices, and optoelectronic devices that perform the transport function, but do not demultiplex signals beyond the wavelength level. The network may contain wavelength-selective switches, ROADMs, optical mux/demux equipment, optical amplifiers, and/or reconfigurable optical cross-connect devices.

It is assumed that the network is designed to prevent a path computation function from configuring non-operable combinations of optical components, and to assure that the addition or removal of an optical signal on a path will not render any of the existing signals in the network non-viable. This may be achieved by engineering individual components for worst-case applications, by imposing constraints on the recognized and allowed topology available for path computation, or by other means.

The path selection functions are assumed to reside in the controller and are not in the scope of this document.

All active devices are assumed to implement the control-data plane interface (CDPI), and all are controlled by a common SDN controller. (For simplicity, server control channels are not shown in the drawing.)

Discussion of allowed overhead, supervisory, or auxiliary channels through the optical network is not included in this use case.

2.2 OPERATION

This section describes the operation of the SDN controller and network elements for different actions within the use case.

2.2.1 Provisioning

When connectivity or capacity demands from the data centers change, the SDN controller is instructed to add or reduce capacity between data centers.

The controller may do this in several ways:

- Redirecting a wavelength from a given source to a different destination.
- Re-tuning a transmitter to a new wavelength (while avoiding disruption to other circuits*) in order to establish a new connection to a different destination.
- Activating a new connection by turning up a previously dark transmitter to add capacity.
- De-activating a connection by turning down a currently active transmitter to reduce capacity.

Implementing this action may require the SDN controller to make connectivity or parameter changes in a number of network elements.

Because the network engineering rules ensure that only optically viable paths are selected, the issue of excessive impairment does not arise. However, minor adjustments such as power balancing are still carried out as appropriate. Any such adjustments must ensure that the parameters remain within specified ranges that enable viability. These adjustments may be calculated either in the equipment itself or through the SDN controller.

2.2.2 Recovery

When a connection fails, the SDN controller must perform the actions required to maintain the desired service level. The transport nodes must notify the SDN controller of the failure of a connection within the time required so that the SDN controller can take the appropriate action. These actions may include notification to the client or establishment of a new connection. The allocation of responsibility and actions taken depends on the maximum time that a service may be disrupted before recovery and overall availability. Within the use case there are three possibilities, depending on the type of failure and the service being supported:

- The desired service level is such that it is acceptable for the connection to be recovered by normal maintenance/repair actions. Depending on the location and type of failure, recovery may take hours to days.
- The desired service level is such that it is acceptable for the connection to be recovered by the SDN controller establishing a new connection to replace the failed connection.
- The desired service level is such that it is desirable for the recovery action to be delegated to autonomous protection/restoration mechanisms within the transport network. In this case, when the original connection is set up, it must include the configuration of the required protection/restoration

* Tuning while active may disrupt other active connections. Re-tuning must be a multi-step process: 1) turn down; 2) re-tune; and 3) turn up.

resources. If recovery action is enabled in both server and client layers, the initial connection setup must include the information required to avoid a single failure in the server layer causing transient switching in the client layer.

- When the working or protection entity fails, the SDN controller must be notified. Depending on the desired service level and the expected repair time, the SDN controller may set up a new protection/recovery path.
- If both the working and protection entities fail, the SDN controller may establish a new connection to recover the service.

2.2.3 Monitoring

Monitoring of the optical layer may be performed to support proactive maintenance, to support fault localization, or as input to SLA assurance.

The degradation of an optical path may at some point cause a failure in the connection that is using that path. The SDN controller may have the capability to compensate for the degradation and must have the capability to issue notifications of any degradation and provide a report on request. This may involve the analysis of parameters retrieved from (or reported by) one or more network elements.

If a network element is capable of detecting the degradation of an optical path, it must notify the SDN controller when the degradation exceeds a preset threshold. This threshold may be preset within the equipment or it may be configured during installation or operation.

If a network element supports monitoring of individual optical parameters, or sets of parameters, the network element must notify the SDN controller when any of the parameters exceeds a predefined range. The range may be preset within the equipment or it may be configured during installation or operation. Parameters that may be monitored include transmitted optical power level, received optical power level.

2.2.4 Network Inventory

The SDN controller must be able to retrieve or be provided with information about the network that it is controlling.

Equipment inventory: A network element must be able to provide information related to the hardware that has been deployed and is under the authority of the SDN controller. This may include, but is not limited to:

- Sub-elements that may be present within the network element (e.g. shelves and plug-in units).
- Capabilities that are supported (e.g. client/server adaptations, application codes, wavelength range, transmitted and received min/max power levels, modulation formats, FEC type, service supported, etc.).
- Component usage state (e.g. active, idle, busy).

Connectivity inventory: A network element must be able to provide a list of the connections that it is currently supporting, if it supports local state information of connections,

Link characteristics: These include, but are not limited to, link loss, distance, and latency. These may be provided either by the network element or specialized test equipment, or configured directly in the SDN controller.

Network topology: The SDN controller must have full knowledge of the network topology, including hardware connectivity. This knowledge may be provided directly to the controller, or by retrieving the results of any discovery protocols that are supported by the network elements.

2.3 BENEFITS

In the absence of dynamic optical services being available from local providers, Transport SDN would enable enterprise IT departments to function as their own service providers, managing their connectivity

and allocating bandwidth between sites in minutes rather than waiting for the local provider to fulfill a fixed-line change order. With Transport SDN, enterprises can switch wavelengths between locations to allocate bandwidth among sites as needed by their applications.

3 Use Case 2: Carrier Ethernet Network Virtualization

This use case illustrates some fundamental aspects of network virtualization in a Carrier Ethernet application—a MEF-6.16.1 E-line service[1]. Crucial aspects of this use case include the recognition of business or organizational boundaries, along with information hiding and namespace translation. As well as static establishment, the use case includes a few illustrative examples (not a complete list) of the kinds of activities that a virtual network client can expect to be able to perform.

3.1 ENVIRONMENT

Figure 2 illustrates a simple physical network, comprising four network elements designated NEs 1–4. In this use case, NE1 is represented as having *m* physical ports on its east side, while NE 4 has *n* physical ports on its west side, identified by <NE>.<ordinal><E/W>. Port 1.1E is assumed to be an uncolored GbE or 10GE optical port; the nature of other physical ports is shown in Figure 2, also only as examples.

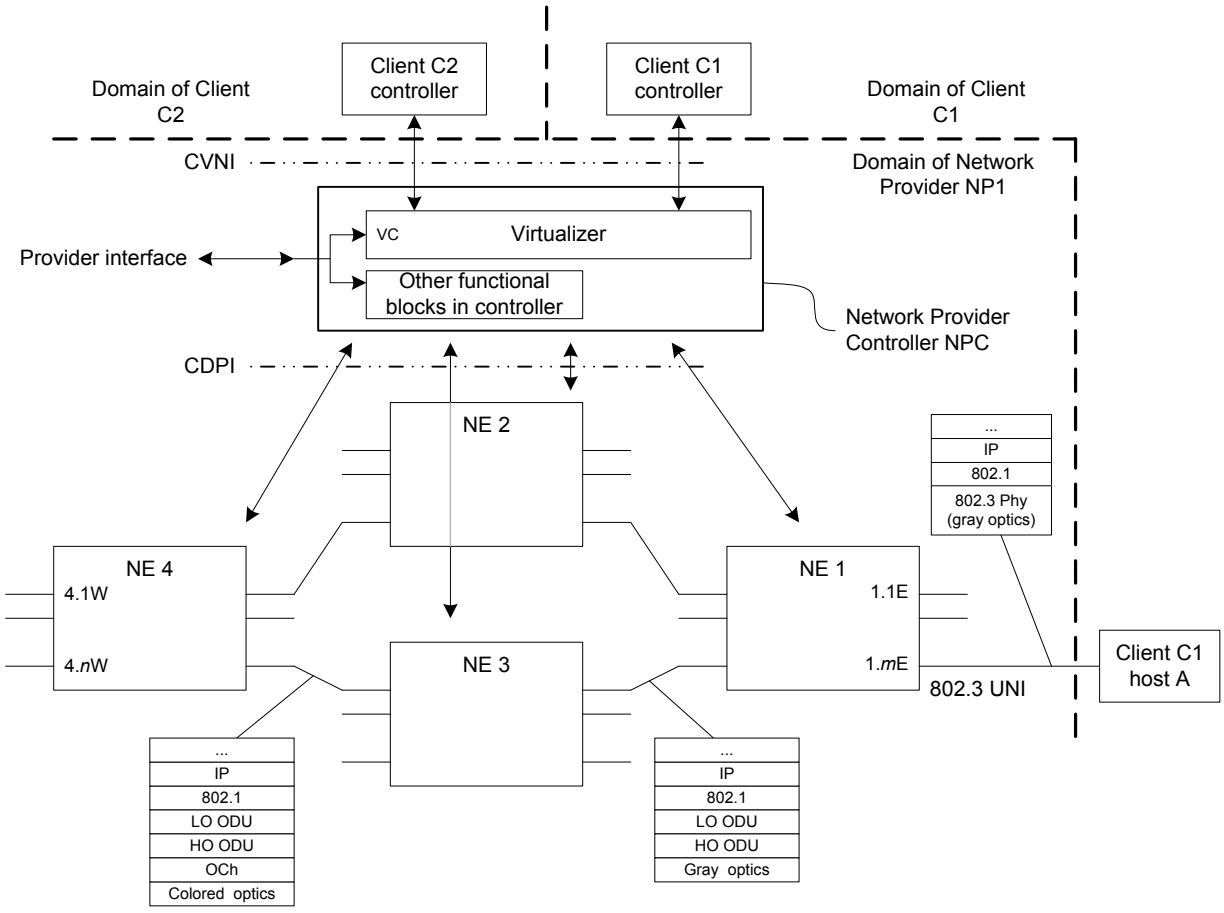


Figure 2 – Physical network, single provider domain

As indicated by the heavy dashed line, the NEs and the network provider's controller (NPC) are owned and operated by a single business entity, Network Provider 1. NP1 manages the network by way of a provider interface, which includes both management of the network itself and an interface VC for

configuration and management of the virtualizer function. Conventional or evolved EMS/NMS systems may use the provider interface, and the NPC may be transparent or semi-transparent to EMS/NMS flows. For purposes of this description, all control and management entities that may appear at the provider interface are understood to be coordinated in their views of network information and state. The provider interface is not expected to be standardized.

Controller NPC has full visibility of and unrestricted access to all NEs. This use case does not distinguish between OF-switch and OF-config. The north and south interfaces from the NPC are called CVNI and CDPI, as shown.

Through a virtualizer function, NP1 offers virtual network services to clients, for example C1 and C2. NP1, C1, and C2 represent separate business entities. NPC therefore represents and enforces a business boundary, which includes functions such as security (AAA), independence between client VNs, information hiding, namespace translation, and address space isolation via encapsulation. The virtualizer ensures that C1 and C2 can neither see nor affect each other.

The virtualizer control interface VC is a trusted interface owned by NP1. There is no business boundary, and information and control are fully transparent across this interface. VC is the means by which the business entity NP1 controls the virtualization functions of NPC.

Figure 2 also illustrates greatly simplified protocol stacks that may be visible at various interfaces. The UNI at port 1.1E, for example, is assumed to be a GbE or 10GE Ethernet private line service that matches the service request from Client 1 for host A's connection. Ethernet services may be transported in the network using a variety of technologies; the protocol stacks shown at the NE 1/3 and NE 3/4 NNIs are one possible example. Other possibilities such as MPLS, PW encapsulation, or additional TDM encapsulations are omitted.

NE 1 is an edge node that adapts customer traffic onto a single-wavelength OTN. At NE 3, the signal may be further multiplexed into an OTN wavelength (optical channel OCh), which ultimately hands off the traffic at NE 4, where there may be another UNI or an NNI into the domain of another network provider NP2.

3.1.1 Preliminaries

This description is written around activities taken by Client 1 by way of its SDN controller. C2 could do the same or different things; two clients are shown to illustrate generality.

Before anything can transpire across the CVNI boundary between the C1 controller and the NPC, each party must establish whatever security is contractually deemed appropriate with regard to the other. Security considerations are important, but are beyond the scope of this use case.

Client 1’s host computers attach to NP1’s network through physical media. Either party may or may not deem the physical plant to be tamper-proof. This point is pursued further in the Operations section.

In general, Client 1’s SDN controller will communicate with the NPC over arbitrary physical channels, so physical security can rarely be assured. At a minimum, the NPC must be provisioned with C1’s credentials. A prudent client controller will also insist on having the NPC’s credentials. Both sides must also be provisioned with security policies; for example, the encryption to be used across the CVNI, the number of parallel open sessions allowed, and idle session timeout.

Before network operations can begin across the CVNI boundary between the C1 SDN controller and the NPC, they must share a common view of the virtual network. Establishing interface points of presence (POPs) is at best a contractual negotiation, and at worst may also require the installation of fiber and equipment. Once the POPs are agreed on, NP1 uses interface VC to configure the virtual network into the NPC, which is responsible for exposing the contracted virtual network to the client. Figure 3 shows the simplest possible virtual network, a set of distributed ports surrounding a single virtual network element.

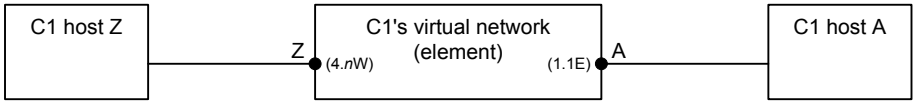


Figure 3 – C1's virtual network

The view exported to C1 must be in terms of identifiers (names) known to C1, in this case, ports A and Z. Associating these identifiers with ports in NP1's network is the responsibility of the virtualizer as configured via VC. In this use case, C1 port A is associated with NP1's port 1.1E; C1 port Z is associated with NP1's port 4.*n*W.

It is possible that NP1 proposes identifiers (possibly auto-generated) and the client simply accepts them. An example of such names for Client 1 might be C1:A, C1:B, etc. It is also possible for the client to specify the names (for example A, B, etc.), which would then be accepted by NP1. In either case, the NPC must maintain mapping between the client's names and its own resources. In this use case, for example, the client name C1:A (or just A, in the known context of Client 1) maps to port NP1:1.1E. Transactions between C1 and NP1 across the CVNI on C1's virtual network occur only in C1's namespace.

This simplest possible use case exposes only POPs to the client, with no details of their interconnection topology. Connectivity, bandwidth, and SLA commitments have been negotiated statically.

As seen in the following section, there are several aspects to the configuration of a virtual network beyond a simple view of topology and connectivity:

- Agreement on a client namespace, alluded to above.
- Because VN client address spaces may overlap with each other, and may also overlap with the provider address space, encapsulation (address mapping) may be required for address space isolation.
- Configuration of policy; i.e., what the client is and is not allowed to do.

If Client 1 were to subcontract network services to Client 11, it would have its own equivalent of VC and NPC, but remain entirely within the constraints of the virtual network subcontracted from NP1.

3.2 OPERATION

In this use case, Client 1 purchases a MEF-6.1 E-line service, chosen as a real-world example of Carrier Ethernet service. Figure 4 shows NP1's view of the resulting service as a heavy dotted line.

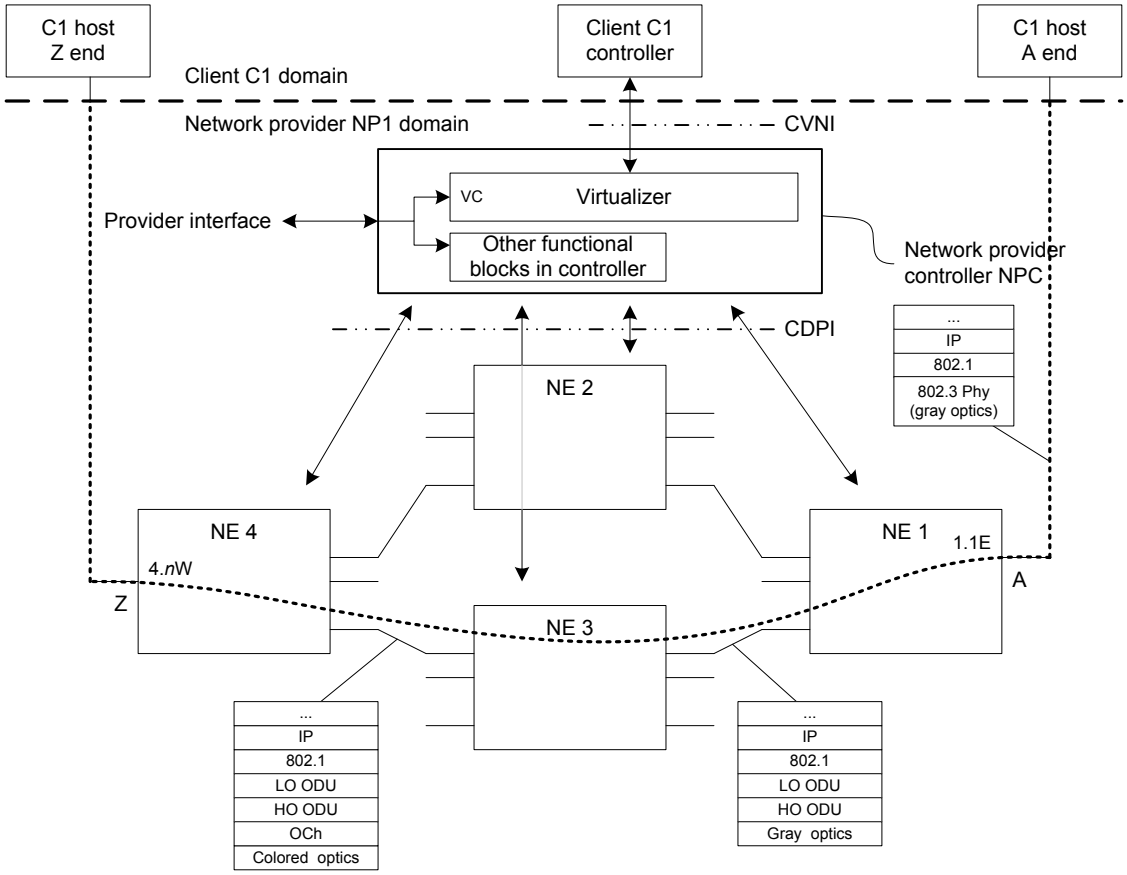


Figure 4 – NP1's view of C1's service

Via VC, NP1 establishes the policies that allow C1 to view and control functions within its virtual network, several examples of which are illustrated below. The examples are intended to illustrate the scope of the necessary functionality, but are neither all-encompassing nor mandatory.

- E-line service to be created between ports C1:A == NP1:1.1E and C1:Z == NP1:4.*n*W, with given bandwidth, delay, jitter, and availability characteristics. (See MEF 10.2 for the list of service attributes[2].) Namespace agreement is described above.
- On request from C1, the NPC computes and sets up the path between ports 1.1E and 4.*n*W, satisfying the contractually negotiated SLA criteria. The NPC may invoke the services of a PCE. Figure 4 shows the physical network as an OTN with colored and gray optics as an example, but multiple layers may be used to transport the service connection (at the discretion of NP1) from wavelengths, to TDM, to L2 and L2.5 forwarding.
- The NPC configures the NEs to police and/or shape traffic from C1's hosts A and Z, as and if required for SLA enforcement. NP1 is responsible for providing the appropriate mapping/translation between the client, NP, and physical network resource names. The mapping/translation function allows the support of independent and possibly overlapping namespaces. Traffic from different clients must be isolated in the transport network. The NPC configures the NEs to provide this isolation using mechanisms appropriate to the core network technology (e.g. wavelength, timeslot or packet assignment, or encapsulation). The method of isolation and its attributes are invisible to the client. Operational security: this use case is an Ethernet service; C1 may request that NPC instantiate an 802.1X client[3] on Ports A and Z to control access from its Hosts A and Z. This is not to be confused with security between the NPC and the client controller C1 itself. If NPC offers this service, there may also be a need to direct the resulting radius messages to an AAA server somewhere else in C1's virtual network, or NPC may host the AAA function itself as an added-value service.

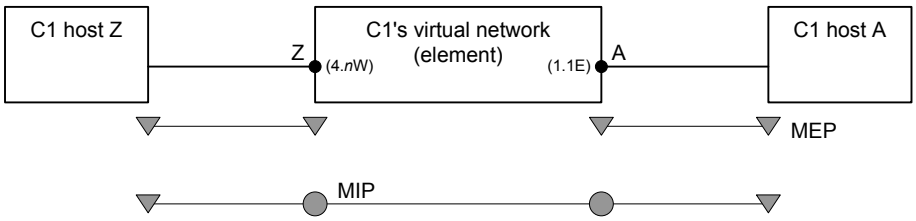


Figure 5 – 802.1ag CFM, C1's view

- Figure 5 shows that C1 may choose to instantiate an IEEE 802.1ag MEP[4] from Ports A and Z to its locally connected host equipment. C1 also instantiates a MEP on its host equipment at either end to confirm end-to-end connectivity. While the host MEPs are beyond the scope of NPC, MIP behavior is required at ports 1.1E and 4.*n*W. Both MEPs and MIPs are provisioned by Client 1 via NPC, using C1's names A and Z.
- C1 configures its local MEPs, those between C1's VN and its hosts, to generate CCMs at 10ms intervals. For this use case, suppose that, if C1 tried to specify 3.3ms intervals, NPC would deny the request, because NP1 has installed a policy to protect its resources from overload. On the other hand, the configuration of C1's end-end (host-based) MEPs to generate CCMs at 3.3ms (or in any other way) is of no concern to NP1.
- Suppose there is a failure or power loss between NP1's port 1.1E and the client. Incoming loss of signal from C1 host A is flagged by NE 1 as an LOS notification against port 1.1E, and reported by NPC to C1 as a notification against port A.
- As to alarm treatment, there are two extremes of contractual agreement. At one extreme, the NPC may provision NE 1 to report a critical alarm, which will cause NP1 personnel to respond immediately. The NPC would probably also forward a critical alarm to C1. At the other extreme, C1 may have committed contractually to the intentional power-down of its network from time to time, in which case NE 1 may only report a non-alarmed LOS event, and the NPC may only forward a non-alarmed event.
- If C1 intentionally shuts down a critically monitored interface, it can suppress spurious alarms by administratively locking port A. This also blocks service on port A. It is a matter of provider policy whether this action also suppresses port 1.1E notifications to the NPC.
- Client 1 can also suppress port A alarms without affecting service through alarm severity settings (change to not-reported). It is a matter of provider policy whether NE 1 continues to notify the NPC about events on port 1.1E.
- C1 requests PM collection via its interface into the NPC. Request and results are expressed in terms of C1's names, A and Z. C1 configures thresholds on the PM and expects to receive notifications when the thresholds are crossed, also in terms of the names A and Z.
- To monitor SLA compliance, NP1 may establish internal PM collection, the same data and thresholds as collected by C1, or different ones. If data or thresholds are different, the NPC is responsible for converting the underlying data to C1's view.
- C1 expects notifications of events that affect its service (for example, unrecoverable failures inside NP1's network) to be presented in terms of its port identifiers (names) and SLA. C1 chooses which notifications to subscribe to, within the set allowed by NP1's policy as provisioned via interface VC.
- To resolve trouble calls, NP1 will also wish to see PM and notifications in the form reported to C1.
- Port 1.1E is dedicated to C1. In addition to being able to administratively lock the port, C1 is allowed to set and retrieve parameters specific to the physical port, such as FEC choices or received optical power level.

- Given that, by hypothesis, C1 only subscribed to an availability figure, and does not have visibility of protection, NP1 is responsible for failure recovery through protection switching or re-routing. C1 cannot take steps to recover from network failures because C1 has non-redundant POPs at A and Z, and no topology view. NP1 notifies C1 only if the SLA cannot be satisfied after a failure. It would also be possible for C1 to have some visibility and control of protection (for example, at L2 via LAGged POPs) with physical protection that remains in NP1's domain. Physical protection visibility and control may be offered to C1 if C1 contracts for full ownership of physical links. (Shared resources cannot be controlled by more than one client.) Intermediate levels may be appropriate when more virtual network topology is exposed.

The NPC has a set of responsibilities:

- NP1 configures internal functions such as OTN trail trace or 802.1ag/BFD CFM. NP1 monitors receive optical power or other measures of SNR, such as uncorrectable FEC blocks.
- The NPC interprets commands, responses, and notifications between C1's names and NP1's resource designations.
- The NPC maps commands, responses, and notifications between C1's virtual network view and the underlying network resources. A command at CVNI may expand into a number of commands at the server layer, directed to a number of different physical equipment units and links, while northbound information may need to be consolidated before being presented to the client.
- The NPC ensures that traffic is isolated between clients, even if they use the same addresses internally—e.g., VLAN IDs or IP addresses. If the handoff points from C1 to NP1 were TDM or wavelengths, with no internal packet visibility throughout NP1's network, isolation could be performed at the level of TDM or wavelength switching. In packet space, this is achieved with encapsulation.
- If C1's controller is an OpenFlow controller, it functions in C1's address space, and does not know about encapsulation. The NPC's encapsulation/decapsulation responsibility therefore includes adding/stripping outer addresses that are passed between the physical network and C1 (OpenFlow commands, responses, and packet-in, packet-out messages). That is, classification rules in the physical switch may contain match fields unknown to C1, but supplied by NPC; likewise with actions for encapsulating outgoing packets. Conversely, packets forwarded from the switch to C1 will have their outermost encapsulation removed by NPC.

3.2.1 Future Work

Although it does push some boundaries of what is reasonable, this use case is intended to introduce virtualization in the simplest possible terms. Further use cases could be written around scope extensions in several dimensions. Examples include:

- More complex services—e.g., multipoint, L2.5, L3. Ability for C1 to set up connections or define SLAs dynamically on particular connections, still working in its own namespace, either using circuit connections or by way of packet forwarding.
- Learning in the packet world, L2 and L3.
- Shared resources at the POPs, with the need to specify some or all of wavelength, high- and low-order TDM, PW encapsulation, VLAN tagging, etc., as selectors of logical ports rather than physical ports.
- C1 subcontracts its view of the network to Client 11, Client 12, etc. In such a case, C1 then enforces a business boundary with its own clients, and has its own virtualizer controller and network provider controller, all bounded by the scope of the virtual network it has contracted with Network Provider 1.
- Network Provider 2 interfaces with NP1 at some set of NNI ports, and a client contracts for a service that crosses the boundary. In some cases, handoff between providers would be negotiated between providers in a subcontractor relationship, and the client would see a single service and billing interface.

3.3 BENEFITS

This use case highlights the highly flexible network virtualization achieved through the use of SDN in control of transport network elements and services, in this case Carrier Ethernet services as represented by a MEF E-Line service. Network resources can be allocated to multiple clients and controlled by each client controller as if it was communicating directly to network elements in a dedicated domain.

A virtual network must be predefined before a client and a server can request or provide service. This is probably a manual process that involves technical and geographic factors and business agreements. In such a case, (virtual) network discovery is a process of confirming interface and topology structures that have been agreed contractually and provisioned (semi-)manually.

However, anonymous and interchangeable resources internal to a virtual network, such as servers in a data center, have no static external interface definitions and may be created, deleted, and discovered.

It is for further study how a client with an existing business relationship could attach to a network, possibly dynamically. Wireless roaming use cases and technology would be at least the starting point for this work.

A physical switch may need to support at least one form of encapsulation to cater for arbitrary client address spaces. A physical switch may need to terminate a virtual flow (classification on outer address) and then forward the contents of that flow based on inner addresses. Physical switches supporting such virtual switch or router functionality may need the ability to logically pass traffic through a classifier-forwarder multiple times.

4 Use Case 3: Optical Network Service Provider Data Center Interconnection (DCI)

This use case addresses data center interconnection (DCI), serving traffic flows between separate data centers (DCs) over an optical transport network. One of the main characteristics of this use case is that the provider network is shared with other client traffic. Another characteristic of this use case is the control separation of the DC and provider network.

DCI control refers to control of networking beyond a DC—that is, an orchestration above DCs. It coordinates with the transport network, allowing coordination of connections between different DCs and across multiple applications.

This use case recognizes differentiation between ownership of DC and network resources to allow different optimizations. For example, the DC and the network could be owned by the same provider, giving the ability to control both DC applications and network resources, e.g., scheduling to maximize utilization. In another example, the DC and the network could be owned by different organizations (e.g., private and public) or a mix (e.g., cloud bursting, the dynamic deployment of a software application that runs on internal organizational compute resources to a public cloud to address a spike in demand). Optimization opportunities in this case would be more limited.

There are two sub-use cases:

- Case 1: The network provider's data centers are interconnected via its own transport network. The separation of DC and network control is a consequence of corporate organization or skill sets, but the network controller trusts the DC controller.
- Case 2: Third-party data centers are interconnected via the network provider's transport network.

For case 1, the DC controller is an internal client to the network provider controller (NPC); for case 2, the DC controller is an external client to the NPC.

Because case 2 requires tighter control in terms of policy, security, and the degree of information sharing between the DCs and the network, discussions are based on case 2 unless stated otherwise.

This use case complements Use Case 2 (UC2). While UC2 describes the concept of network virtualization and its components in general settings, the focus of this use case is to contextualize the

concept described in UC2 into a data center interconnection setting, and to provide concrete workflows needed between the NPC and DC controller for virtual network service establishment, virtual network connection, and fault monitoring, detection, and recovery. Specifically, this use case provides an automated process of verifying interface and topology structures that have been agreed contractually between the NPC and the DC controller. This use case also demonstrates the use of dynamic/automated reconfiguration of a virtual network service, which may, for example, be required following contractual changes to the service.

4.1 ENVIRONMENT

Current DCI is based on pre-allocated static WAN optical pipes between DC sites. This pre-allocated capacity may be engineered for peak rates and thus underutilized much of the time due to fluctuating traffic demands. This mode of operation is not suited to dynamically allocating new applications to one out of a number of candidate DC sites while adjusting the WAN bandwidth accordingly. For example, some workloads or data may need to be migrated on the fly from one data center to another. Disaster recovery is another example in which a large amount of data may need to find alternative data centers when the currently serving data center experiences an outage affecting application performance.

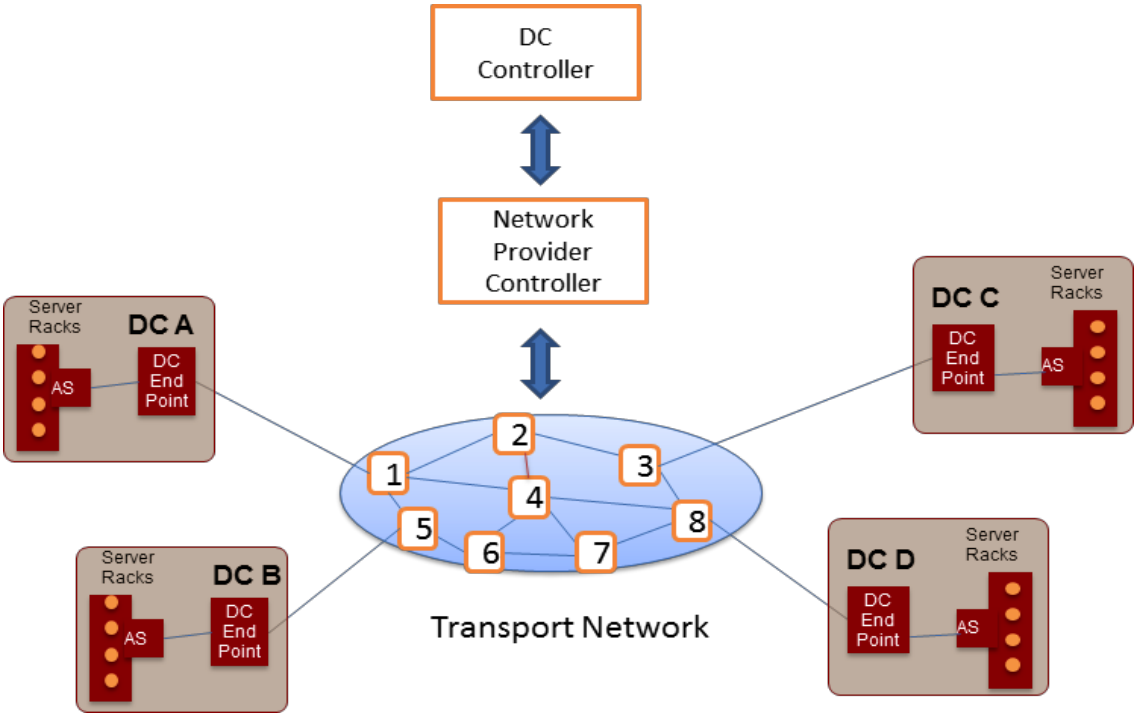


Figure 6 – Data center interconnection architecture

Figure 6 depicts a high-level network architectural context for this use case. The service provider's transport network may support a number of different client applications, including DC interconnection.

There are a few assumptions:

- The DC controller knows all its DC endpoint interfaces that are connected to the provider network.
- A data plane connection between each DC endpoint interface and a corresponding network provider endpoint interface (i.e., UNI) is assumed to have been established prior to controller communications between the DC controller and the NPC. (This assumption may be replaced by dynamic establishment of a data plane connection in a use case that supports dynamic attachment to the provider network, e.g., via wireless access technologies.)

- A service contract is in place between the DC operator and the service provider that sets the relevant policies regarding the operation of the service(s) available to the DC operator (and by extension to the DC controller).
- The NPC knows the provider network endpoint interfaces that are connected to DCs operated by the DC operator and covered by the service contract. (This assumption may be replaced by an authentication mechanism in a use case that supports dynamic attachment to the provider network, e.g., via wireless access technologies.)
- The DC controller has full visibility of each data center under its control. This visibility includes DC resource, DC location information, interfaces to provider networks, and other user/application-related information.
- Multi-layer or packet optical integration (POI) aspects of transport networks are covered in Use Case 4 of this document.

For the DC interconnection application, the client controller is the DC controller, which can be an internal entity or an external entity with respect to the relationship with the service provider. Each data center may have a local DC controller, and these DC controllers may form a confederacy or hierarchy to interface the NPC. How these DC controllers are organized to present a unified interface to the NPC is beyond the scope of this document. For purposes of this use case, a single logical DC controller is assumed to connect to a single logical NPC.

4.2 OPERATION

Three use case scenarios and workflows are described in this section.

- Virtual network service establishment
- Virtual network connection and virtual network element connection
- Fault monitoring, detection, and recovery

4.2.1 Virtual Network Service Establishment

This use case scenario is concerned about pre-virtual network connection information exchange and its workflow. This process involves negotiation of virtual network service (VNS) between the DC controller and the NPC within the confines of the established business agreement. This process meets two objectives: to verify the established business agreement between the DC operator and the service provider, and to automate business negotiation/re-negotiation and the virtual network service (VNS) creation processes.

A customer defines the need for a virtual network within the provider's transport network, then contacts the service provider to negotiate a contract for VNS. The customer provides a traffic demand matrix as part its VNS request, and the service provider computes and presents to its customer one or more virtual networks (VNs) that would support the requested objective, including bandwidth. Each VN consists of one or more virtual network elements (VNEs), which are interconnected by virtual links. These links can be characterized by a number of attributes, including committed bandwidth, excess bandwidth, latency, maximum number of supported connections, etc. (See Appendix A for VN examples.)

The negotiation of VNS is per VNS instance. The VNS must be instantiated before the first virtual network connection is to be set up. The instantiation of the VNS will result in the allocation of the committed resources of the VN. The resources in the network path in the subnetwork (e.g., cross connect resources) are only allocated when the VNE connection setup is performed.

4.2.1.1 Negotiation initiation

This description is written around negotiation between the DC controller and the NPC. The actor that initiates negotiation is the DC controller. As a client to the transport network, the DC controller is interested in knowing relevant transport network resource information, especially in light of DC

interconnection. Initially, the DC controller must negotiate with the NPC to identify the set of endpoints it wishes to connect. As part of the negotiation, the DC controller may also express traffic characteristics that need to be supported between a set of endpoints such as traffic demand (i.e., bandwidth), and QoS requirements associated with DC endpoint interface pairs.

To allow negotiation to take place, the correspondence between DC endpoint interface identifiers (DC EPIDs) and provider network endpoint interface identifiers (PN EPIDs) must be established. This may be accomplished using a manual process (e.g., exchanging identifiers between DC operations and PN operations personnel), or it may be automated, e.g., using LLDP[5] to exchange endpoint identifiers at the user-network interfaces [UNIs]). If the endpoint interfaces are under the control of OpenFlow SDN, this exchange can be done using PACKET_OUT and PACKET_IN messages. By this exchange, both the DC controller and the NPC can acquire the association between the DC EPID and the PN EPID. Usage of protocols such as LLDP or OF requires some authentication of the exchanged information or of the endpoints exchanging the information.

4.2.1.2 Negotiation response

During virtual network service negotiation with the DC controller, the NPC is the actor for creating a response to the DC controller with an associated VN.

If the DC endpoints are identified in the CVNI using the DC EPID, the NPC must translate each of these to the associated PN EPID before proceeding to process the request. If the endpoints are identified by the DC controller using the ID pair (DC EPID, PN EPID), the NPC need not access a translation service and can process the request using its own identifier from each ID pair.

The NPC provides a virtual network (VN) in response to the DC controller. The granularity of the VN has been pre-negotiated by the mutual business contract and policy and is expressed in terms of virtual network elements (VNEs) and virtual links and their identifiers. See the formal definition of VNE and virtual link. In order to provide a relevant VN, the NPC initiates a request to the virtual network computation entity of the NPC and determines the feasibility of the request. The corresponding result will need to be sent by the NPC to the DC controller. The algorithm to compute a virtual network is out of scope. The implementation of the virtual network computation entity within the NP controller is also out of scope.

4.2.1.3 Negotiation initiation/response information

The following elements and associated parameters should be supported at a minimum in the VNS negotiation initiation/response:

- **VNS instance identifier.** This identifier identifies this particular instance of VNS.
- **Traffic matrix element.** This element describes traffic demand and other QoS information associated with DC endpoint interface pairs (e.g., latency). The connectivity type, bandwidth type, and directionality are a part of the traffic matrix element.
    - Connectivity type:
        - Point-to-point
        - Point-to-multipoint (future consideration)
        - Multipoint-to-multipoint (aka “anycast”) (future consideration)
        - Rooted multipoint (future consideration)
    - Bandwidth type:
        - Committed bandwidth resources
        - Excess bandwidth resources/shared pool (future consideration)
        - Others (future consideration)

- Directionality of each connectivity type:
  - Unidirectional
  - Bidirectional
- Location information element. This element describes the DC endpoint interfaces associated with the connectivity element. For unidirectional connectivity, the source list and the destination list need to be distinguished.

- **VN description element.** This element describes the VNEs and virtual links and their identifiers that belong to the VN.

4.2.1.4 Virtual network creation

Following negotiation of the virtual network service, the NPC allocates network resources needed to fulfill the agreed upon VN. When these resources have been allocated, the NPC notifies the DC controller of the availability of the VN for transport of information between DC endpoints. The resources in the network path in the subnetwork (e.g., cross-connect resources) are only allocated when the VNE connection setup is performed, which is described in the subsequent section.

4.2.2 Virtual Network Connection

When the DC controller receives a service request from one of its business applications or from its network engineering function, the DC controller's path computation function computes a path through the virtual network based on the VNS establishment process. After path computation, the DC controller will issue VNE connection setup commands and send these commands via the CVNI to the NPC.

4.2.2.1 Connection command

The actor of the VNE connection command is the DC controller.

The DCC sends a VNE connection setup command to each VNE in the path computed by the DCC's path computation function. Each VNE sets up the connection in the NE or in the subnetwork represented by the VNE. The DC controller is in full control of the path through its VNEs and virtual links/tunnels. When sending VNE connection commands to each VNE, it is the DCC's responsibility to provide them with relevant parameters.

Any change of existing active VNE connections, such as bandwidth modification, should also be supported. There may be bandwidth modification or other changes to the VN state (e.g., capacity, delay, etc.) that fall outside the constraints of the existing virtual network service agreement and would necessitate a renegotiated agreement. It is expected that renegotiation will be done based on the same VNS instance with the modified attributes. The renegotiation of a completely new VN is for a future consideration.

4.2.2.2 Connection confirmation

The actor of the VNE connection confirmation process is the NPC.

Upon the receipt of a VNE connection setup command from the DC controller, the command is either converted into a command to set up the connection in the NE represented by the VNE, or the NPC's path computation function is triggered on a VNE representing a subnetwork to compute a path through this subnetwork that would comply with the VNE connection setup command. Following path computation on each VNE, the NPC is responsible for the setup of a connection in each NE within the computed path of a subnetwork in the provider network.

In some cases, the VNE connection setup command cannot be completed successfully. A subnetwork outage, for example, may cause a setup command to fail due to insufficient bandwidth. The failure response should include enough information to permit the client to evaluate alternatives and possibly initiate new VNE connection commands. This recovery action should be done within the constraints of the

established SLA. Any recovery action beyond the SLA level may necessitate renegotiation of the virtual network service agreement by the user and service provider.

Multiple pre-negotiated policies may exist between NPC and DC controllers, their terms depending on the virtual network service contract and the relationship of the NPC and DC administrative authorities. For example, in the case that the NPC and the DC controller are operated by the same service provider, more detailed information on the virtual network may be provided by the NPC. On the other hand, when the NPC and the DC controller belong to different administrative entities, hiding network details and additional security measurement are necessary to protect the network from potential attacks.

4.2.2.3 Connection information

The CVNI should support VNE connection setup, modify, and tear-down commands and confirmation mechanisms. The following elements and their associated parameters should be supported at a minimum level:

- **VNE identifier.** This identifier identifies the VNE.
- **VNE connection description element.** This element describes the connectivity type, bandwidth type, directionality, and other parameters for the VNE.
  - Connectivity type:
    - Point-to-point
    - Point-to-multipoint (future consideration)
    - Multipoint-to-multipoint (aka “anycast”) (future consideration)
    - Rooted multipoint (future consideration)
  - Bandwidth:
    - Committed bandwidth resources
    - Excess bandwidth resources/shared pool (future consideration)
    - Others (future consideration)
  - Directionality of each connectivity type:
    - Unidirectional
    - Bidirectional
  - Location information element. This element describes the VNE connection endpoint interfaces (e.g., virtual links, physical links, logical ports) associated with the VNE connection. For unidirectional VNE connections, the source list and the destination list need to be distinguished. For rooted multipoint VNE connections, the root, leaf, and leaf group ports need to be distinguished.
  - Protection/restoration element. This element describes diversity criteria requested/applied for each VNE connection; e.g., diversity requirements relative to other VNE connections.
  - Service duration element. This element specifies the VNE connection duration in terms of begin time and end time. This element is optional; if not present, the end time is undetermined and the begin time is immediately.

4.2.3 Fault Monitoring, Detection, and Recovery

Fault monitoring information can also be exchanged between the DC controller and the NPC. This function could be complementary to SLA management; it is related to fault and performance notifications to the NPC with consequent actions.

We have to consider two types of faults: one associated with the VN resources degrading or failing (partially or entirely), and the other with faults in virtual network connections. If a virtual link is protected or restorable as part of the SLA, then the NPC may hold off any failure/degradation report to the DC controller while it is protecting/restoring the virtual link.

Examples of fault or performance issues affecting SLA may include:

- An irrecoverable fault has occurred in the optical transport network. This could be due to a variety of reasons, such as optical transport network protection/restoration is ineffective due to an unmanaged failure in the backup facility or insufficient/pre-empted resources, or the operator did not deploy survivability mechanisms.
- The optical transport for the DCI connection still exists, but the service is degraded. For example, guaranteed bandwidth cannot be satisfied, required latency cannot be satisfied (may be augmented with current latency value), an increase in BER/EBR occurs, etc.

Responses to fault and performance notifications may include:

- The network provider controller, based upon associated network operator policies, determines to take appropriate consequent actions within the optical transport network (on a case-by-case basis).
- The network provider controller triggers execution (through DC controller) of appropriate consequent action policies for clients/customers.

4.3 BENEFITS

This use case provides many benefits for both DC operators and service providers. Such benefits include improving optical transport network control and management flexibility (for example, the ability to deploy third-party client management/control systems), and the development of new service offerings by network virtualization. The CVNI enables programmatic and virtual control of optical transport networks by allowing applications to have greater visibility of and control over connections carrying their data, and the monitoring and protection of these connections, subject to operator policy.

5 Use Case 4: Packet-Optical Integration

The goal of this use case is to show integration of packet and optical network control. Such integration supports joint optimization for greater efficiency and takes advantage of knowledge of topologies and status across layers, as well as dynamic capabilities supported by the optical transport network.

5.1 ENVIRONMENT

One example of packet and optical network interoperation is a connectionless IP network running packets over an L0/L1 transport network (e.g., OTN), as shown in Figure 7. The IP network makes use of L0/L1 links connecting its routers and runs IP routing protocols based on the Layer 3 topology.

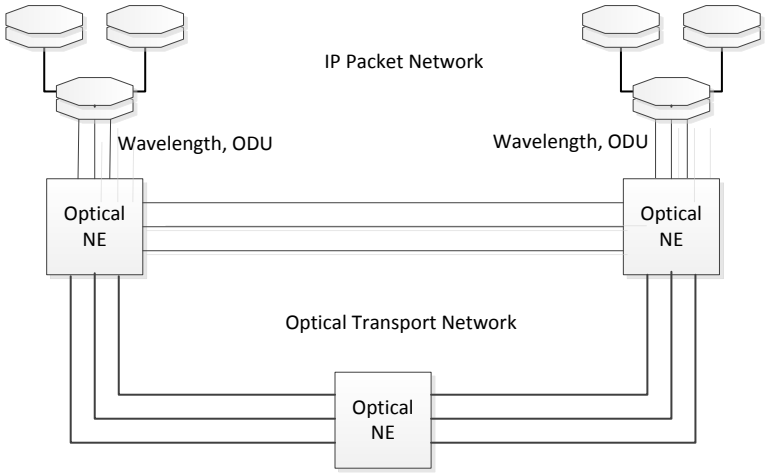


Figure 7 – Example of packet and optical networks

The typical control framework for this is independent control of IP and of optical transport. The optical transport network manager and IP network manager are separate, as shown in Figure 8. Traffic engineering/path computation is done in the IP packet domain without knowledge—or taking advantage—of the optical transport network's current topology and how it maps to the packet topology. Cut-through paths (i.e., bypassing the packet network) that may be available using the optical transport network directly may not be used even if they have lower latency and are more efficient. Both layers implement distributed control planes for forwarding control without multi-layer coordination/integration, making optimized resource utilization a significant challenge.

The optical transport and packet networks may in some cases have limited interaction facilitated by a control plane UNI interface between IP and optical network elements. This requires implementation of an optical UNI (such as the GMPLS or OIF UNI) between router and optical switch supporting connection requests or exchange of information between the packet and optical transport layers. The UNI has limited functionality, as it does not support exchange of topology or resource status information for path computation, nor global optimization and control.

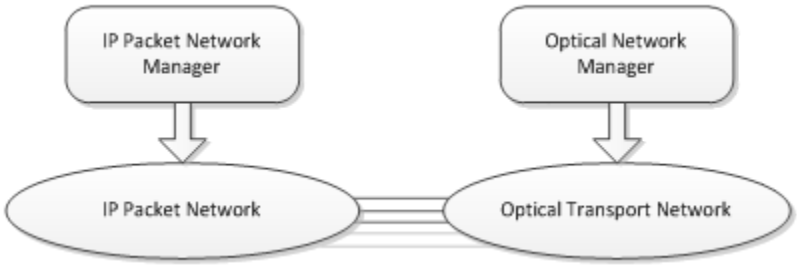


Figure 8 – Independent operation of packet and optical networks

A different example of packet and optical interconnection is shown in Figure 9, where the optical transport network supports converged packet and optical functions, and a packetized transport mechanism such as MPLS-TP is used within the optical transport network for more efficient aggregation and forwarding of IP traffic from the routers. Figure 9 shows the optical NE taking incoming packets and allocating them to different MPLS-TP LSPs, which then are mapped into OTN connections taking a direct or longer path depending on packet characteristics such as address or class of service.

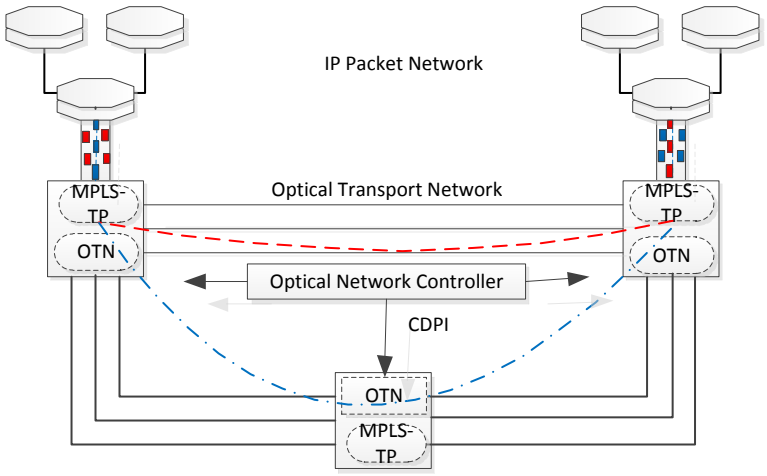


Figure 9 – Optical network with converged packet functions

This scenario has been the subject of previous research on multi-layer SDN[6]. In this case, a multi-layer CDPI control interface would be used between the Optical Network Controller and the network elements to control packet and optical forwarding decisions, using a multi-layer switch abstraction model. The use of integrated packet and optical switching within the transport network enables reduction in equipment costs and more efficient allocation of bandwidth within the network. The multi-layer CDPI interface supports the ability to program packet and optical forwarding within transport network elements that combine packet and optical fabrics.

Both scenarios will be discussed below in terms of the operation of the SDN/OpenFlow control plane.

5.2 OPERATION

This section describes workflows and operation for SDN/OpenFlow-based packet-optical integration, through two potential controller relationships:

- Single controller using multi-layer CDPI for packet/optical control
- Hierarchical controllers using CVNI between client and server controllers, with separate controllers for packet and optical networks

5.2.1 Controller Relationships

OpenFlow-based SDN can be used either with a multi-layer controller or with a hierarchy of controllers and separate controllers for each layer.

In the first scenario, a multi-layer controller controls both optical and packet transport functions, as shown in Figure 10. The multi-layer controller has the ability to set both packet and optical forwarding tables and control the adaptation between layers.

- An integrated multi-layer OF supports both packet and optical switching technologies.
- Both packet and optical layer topologies can be adjusted to fit demand.

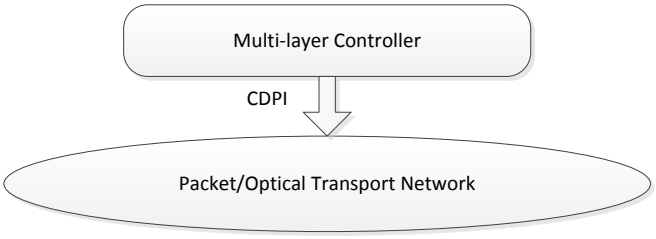


Figure 10: Multi-layer controller

The second scenario is where the network provider controller coordinates separate packet and transport controllers, as shown in Figure 11. The NPC orchestrates IP packet network and optical transport networks, and sees the IP packet network topology together with the optical transport network topology. In this case, the optical network controller (ONC) supports a CVNI to the NPC to support information and control of the transport network. An abstracted view of optical transport network topology is provided by the ONC using the CVNI; this may provide detailed topology or simplified topology based on the requirements of the NPC.

Information is provided via the CVNI in the abstract topology to identify fate-shared risk groups, latency, cost, etc., within the optical transport network in order to support efficient traffic engineering.

An example of basic topology simplification that might be applied is collapsing multiple physical links between nodes into a single TE link supporting the combined total bandwidth, simplifying path computation.

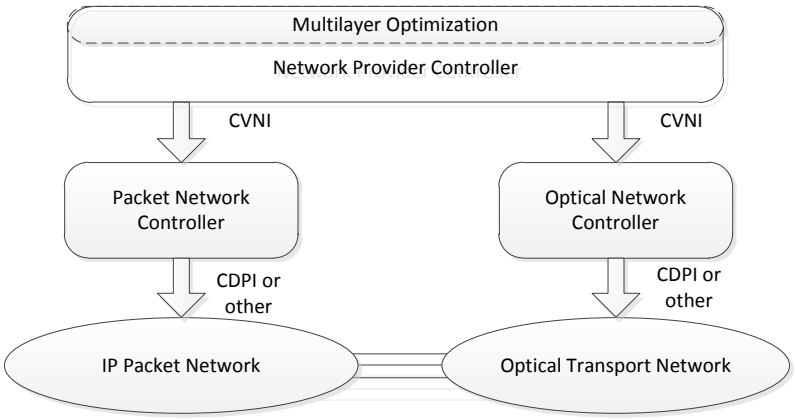


Figure 11 – Multi-layer control with separate packet services and optical network controllers

In this scenario, a distributed control plane can be used within the IP packet or optical transport network, leading to a hybrid SDN environment combining an SDN controller at the provider level and distributed control plane within a network domain. The CVNI interface provides an abstracted view of the network, and is independent of whether the controller below the CVNI uses a CDPI to all network devices, some combination of CDPI and distributed control plane, or a distributed control plane alone.

For example, an operator can use an ASON/GMPLS control plane in the optical transport network for regular operation, but allow the ONC to direct flow/connection set-up for special requests. Alternatively, the operator can use the ONC for global optimization and path computation, and the distributed control plane for connection provisioning. Another option is to use the CDPI to configure forwarding at the ingress

and egress nodes and distributed control plane to control establishment of forwarding paths across the domain from ingress to egress. The model used does not impact the CVNI interface to the NPC.

5.2.2 CDPI Operation for Multi-Layer Control

In the multi-layer controller scenario, a multi-layer CDPI interface can be used to devices that incorporate both packet and optical functionality, for example the edge nodes supporting both MPLS-TP and OTN shown in Figure 9. An information model is essential for interoperability. Figure 12 sketches the model for multilayer CDPI at a very high level.

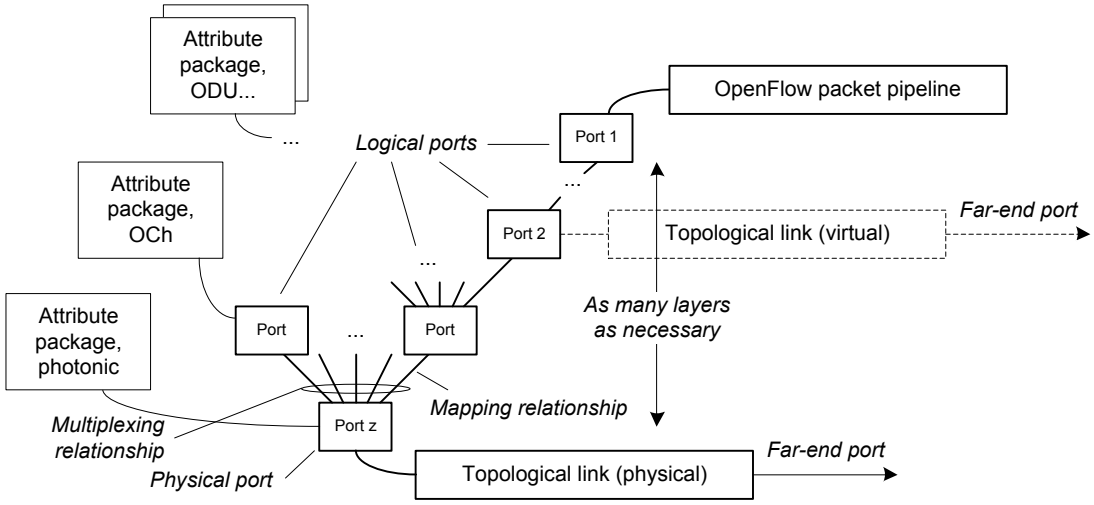


Figure 12 – Information model sketch

Figure 12 illustrates how the OpenFlow packet pipeline terminates in a logical port, which may be one of several ports multiplexed into a lower server layer. Depending on the level of network virtualization exposed to the client, a server layer may be associated with a virtual tunnel to some far-end port (or ports), or it may be further mapped into an even lower server layer. At each mapping layer, the port managed object is associated with a set of attributes and functions appropriate to its level. The bottom of the multiplex stack exposes a physical link, such as a multi-wavelength fiber.

A given configuration contains as many or as few nested ports as are needed to represent the mapping and multiplexing hierarchy.

The following sequence of events illustrates how an ONF SDN controller can bind a given packet flow to a transport network connection in a given logical switch. A logical switch is a resource-isolated subset of a physical switch. Similar steps would be taken in parallel on other switches along the route of the connection.

1. In accordance with the existing OF-switch model, the controller may already know about (logical) Port 1 as the termination of the packet pipeline. In some scenarios, Port 1 will need to be instantiated and bound to the packet pipeline.
2. The controller instantiates a server Port 2 and associates Port 2 with Port 1 through a protocol mapping. Depending on the protocol stack, Port 2 is accompanied by a set of mapping option attribute values and references to supporting functions. Examples of mappings include PW or MPLS-TP tunnel encapsulation or GFP-F encapsulation and mapping into an ODU payload. The attribute packages shown in the Figure 12 also suggest OTN as one of the possible stacks. Supporting function examples could include BFD terminations or auxiliary channels.
3. The controller continues to instantiate server ports for further layers of the stack as necessary. Both mapping and multiplexing are illustrated in the figure.

4. Ultimately, the protocol stack maps to physical Port Z, at which point it enters a physical link to an adjacent device.

At an intermediate network element—i.e., one that does not decompose the flow for separate packet-based forwarding—only the lower layers of information need be instantiated. The connection can be passed through the logical switch.

This example is written around the lowest-level SDN controller, which has unrestricted access to the physical resources. A client SDN controller may operate equivalently on its view of a virtual network which is being provided by a server layer controller. The client controller believes it has full control over resources made visible by the server layer. The client cannot see or control ports that are controlled by other clients or are held in reserve by the network provider. In such a case, the lower layer ports are not visible to the client.. This implies that, from the client's point of view, the stack of ports may stop at some intermediate level, for example at Port 2 in Figure 12. In such a case, the lower layer ports are not visible to the client.

5.2.3 Hierarchical Controller Operation

This description is written around activities taken by the provider's optical network controller (ONC) using a CVNI interface to a network provider controller (NPC) that orchestrates packet and optical transport network actions as in Figure 11.

It is assumed that the service provider determines as part of its internal policy whether abstraction of topology is used in the interface between the NPC and the ONC. Abstraction of topology provides some advantages in reducing the amount of information that needs to be exchanged between controllers and reducing the complexity of path computation and analysis for the NPC, at the cost of some loss of information.

In initial topology exchange, the NPC and ONC synchronize on an abstract topology view of the optical transport network, including the switches in the optical transport network, the ports on each switch, and the connections between ports and their characteristics. A common naming scheme is used to label objects in the presented abstracted network topology; however, the ONC may need to support a mapping from the abstracted topology to the physical network devices and ports. This topology, together with the IP network topology, provides the NPC with the ability to globally optimize traffic distribution across both optical and IP networks.

The CVNI supports passing of abstract network topology information from ONC up to the NPC to provide the basis for path computation and optimization across IP packet network and (virtual) optical transport network

The network provider (as the sole actor) wishes to provision packet flows across the optical transport network. The CVNI has provided topology information during initial topology exchange that allows the NPC to compute the best path for this new flow—for example, using an existing optical transport network connection or creating a new one to carry the packet flow.

Note that an existing optical transport network connection is represented as a link in the IP packet network topology. A new connection in the optical transport network is represented as a new link in the IP packet network topology.

If a new connection is required in the transport network, the network provider uses the CVNI interface to provision connectivity from the corresponding optical transport network ingress port to the corresponding optical transport network egress port in the abstracted topology by means of a series of commands directed towards the virtual network elements within the path. These commands are mapped by the ONC into commands to provision the necessary port/packet matching and egress port selection for the packet flow in the ingress node, the associated matching and port selection for the transport connection in the intermediate transport nodes, and the port/packet matching and egress port selection for the packet flow at the egress node.

At this time, any associated autonomous network functions such as protection or mesh restoration are also provisioned using the CVNI as part of the connection.

The CVNI supports modification to the optical transport network connection set as determined by the NPC (e.g., new connections or change in connection bandwidth), as well as changes to mapping from incoming packets to egress ports at ingress and egress points of the abstracted topology.

5.2.3.1 Reaction to events in the transport network

An event in the transport network such as a failure or maintenance action may affect an associated packet flow. Examples of such transport events include:

- A fault that cannot be immediately corrected, e.g., because the failed link is not protected or the protection path is not available (for services with no fully guaranteed protection).
- A change in performance, e.g., an increase in bit error rate, latency, or guaranteed bandwidth (for flexible bandwidth service).

In this case, the event is reflected to the ONC through the CDPI through, for example, an event indication or change in port statistics. The ONC maps this to any associated indications carried up to the NPC through the CVNI.

The NPC, based on associated network operator policies, determines and executes (via the ONC or the IP packet network controller) appropriate consequent actions. Actions within the optical transport network are triggered via the CVNI to the ONC. This may, for example, involve rerouting of transport network connections. Actions within the IP packet network may also be executed that are not visible to the transport network, e.g., rerouting of packet flows.

Based on any characteristics associated with the transport network connection that require autonomous action by the optical transport network (such as protection or mesh restoration), the transport network connection is recovered and a new indication reflecting any changes to the path of the connection is passed back to the ONC, and through it back to the NPC.

The CVNI supports the ability for the client to mark connections eligible or ineligible for automated rerouting by the server layer, the indication of faults or performance change in the topology passed from the ONC to the NPC, and changes to connections or packet mapping passed from the NPC to the ONC.

5.2.3.2 Re-optimization

Longer-term changes in the transport network topology may be incurred due to failures or to the introduction of new elements into the transport network. Further, the creation and deletion of connections over time may have resulted in non-optimal resource allocation across the optical transport network. Information is fed back to the NPC through the CVNI. The NPC can use this information to re-optimize resource allocation across both IP packet and optical transport networks.

The NPC may find the optimal situation of the network thanks to the new information. However, there would be traffic flows which cannot be moved from the network, while other traffic flows can have controlled traffic cuts during the night. These conditions may lead to a re-optimization process with intermediate steps. This re-optimization process can be automated, but network operators may require human approval in the process. In each step, the network situation should be presented to the operator, who can select whether traffic should be rerouted now or delayed to another time.

The CVNI supports changes to connections or packet mapping passed from the NPC to the ONC, as well as any updates to abstract topology attributes passed from the ONC to the NPC.

5.2.3.3 Reaction to packet traffic changes

This section encompasses and extends the previous sections to consider changes in traffic load.

In this use case, the edge nodes of the IP packet network should provide traffic data to an optimization engine on a continuing and semi–real-time basis. The data is largely performance monitoring (PM), such as a summary of bytes (not packets, because capacity is what we care about) delivered from one end/edge point to another, and classified in any number of ways. Delay and loss PM are also useful.

In the following paragraphs, PM is chosen to collect traffic data and the NPC will host the optimization engine.

This optimization engine also knows about network topology, including intrinsic delay, capacity, and protection. Because this is an SDN-controlled network, the engine is also is kept up to date with changes in the forwarding map (or stabilize connectivity/topology in a global viewpoint).

The optimization engine can re-route traffic at packet services network or optical transport network layers. Its decisions are communicated either to the packet forwarding tables via traditional OpenFlow switch, or via whatever CDPI extensions are appropriate into the optical transport network world, or both.

It is important that the optimization algorithm be constructed to avoid oscillatory behavior.

The CVNI supports changes to connections or packet mapping passed from the NPC to the ONC, as well as reporting of PM data from the ONC to NPC.

5.3 BENEFITS

This use case explores the benefits of OpenFlow/SDN control integration for packet/optical networks. Use of Transport SDN for multi-layer IP/MPLS plus Transport optimization achieves many network benefits, including:

- CapEx reduction, by reducing the need for over-provisioning of the network to support demand shifts and protection/restoration, through the integration of control over packet and optical networks.
- Increased service availability (e.g., coordinated protection and restoration) and service quality (e.g., latency-optimized multi-layer provisioning) based on integrated reactions to changes in network conditions.
- OpEx reduction and simplification through automation to reduce manual processes and associated configuration errors, compared to separate control structures requiring manual coordination.
- Increased revenues, by leveraging network intelligence to monetize the network based on a broad list of programmable path and service level parameters, such as end-to-end latency of packet service.

As shown, OpenFlow/SDN supports multiple approaches to control plane integration, providing flexibility to address different environments faced by the service provider.

Appendix A: Virtual Networks

A virtual network (VN) is a client view of the transport network. It is the view that the network operator provides to the client. It only shows the relevant client endpoints and some level of network connectivity (which depends on the granularity level negotiated between a client and the network), while hiding and condensing the real physical network topology. Let us look at the network topology depicted in Figure A.1 as a base physical network topology of a service provider.

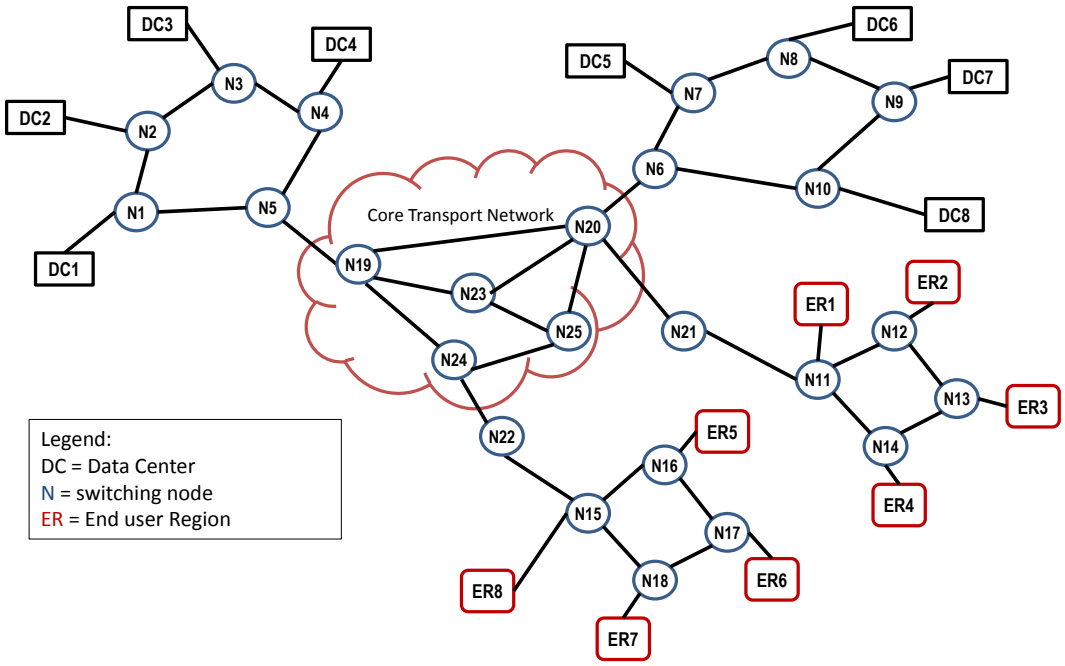
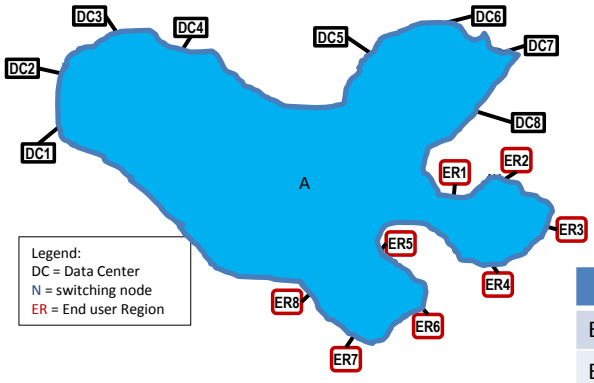


Figure A.1 – Example transport network physical topology

As an illustration, Figure A.1 shows a number of DC locations (DC1, …, DC8), which are accessible by end-user regions (ER1, …, ER8). As depicted in this figure, there is a core transport network in the middle, through which end users access data centers for their applications. It is also possible that traffic may be generated between DC locations for certain applications. This topology is a physical topology.

The next few figures (Figures A.2 to A.4) show different levels of virtual network topology, which are abstracted views of Figure A.1. It is important to understand that if Figure A.1 represents the VN exposed by the server to the client, then all of the less granular views can be constructed entirely within the client's space, for example by a GUI. There is no need to further involve the server in abstracting away detail. If the client were to desire additional detail, however, a new VN would have to be negotiated with the server.



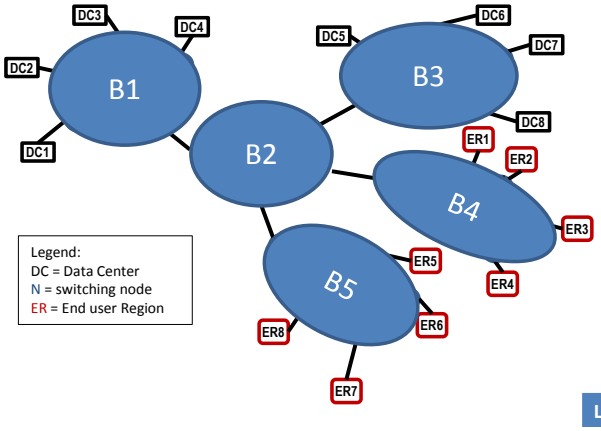
End Point VN View does not give any resource bottleneck information

Minimum Hop (ALTO network cost map)

	DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8
ER1	7	8	8	7	6	7	7	6
ER2	8	9	9	8	7	8	8	7
ER3	9	10	10	9	8	9	9	8
ER4	8	9	9	8	7	8	8	7
ER5	8	9	9	8	9	10	10	9
ER6	9	10	10	9	10	11	11	10
ER7	8	9	9	8	9	10	10	9
ER8	7	8	8	7	8	9	9	8

Figure A.2 – Endpoint topology view, low granularity

In Figure A.2, the entire transport network is abstracted as a single virtual node. This is the least granular abstraction of a physical topology. Connectivity detail is hidden from the view of all endpoints in the domain of interest. In this particular example, the cost between endpoints is a hop count in a physical topology. Port pairs interconnected by tunnels may be represented with a hop count of 1.



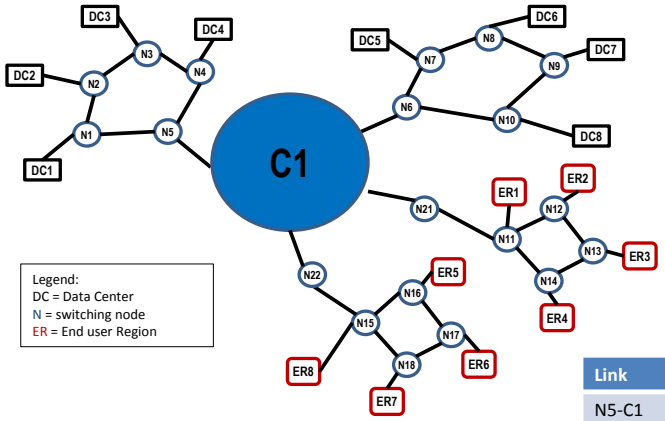
Partial VN View does give the bottleneck Information for some key bottlenecks

Graph Representation

Link	Capacity (Gbps)	Cost
B1-B2	100	1
B2-B3	50	1
B2-B4	50	1
B2-B5	25	1
DC1-B1		
.....		
ER8-B5	10	1

Figure A.3 – Partial graph topology view, medium granularity

Figure A.3 shows a bit more granular view of topology abstraction. In this particular topology, the network is partitioned into five virtual nodes (B1, …, B5), with links among them and the access links from endpoints. This is a graph representation with multiple cost types. This level of VN abstraction gives some resource capacity information.

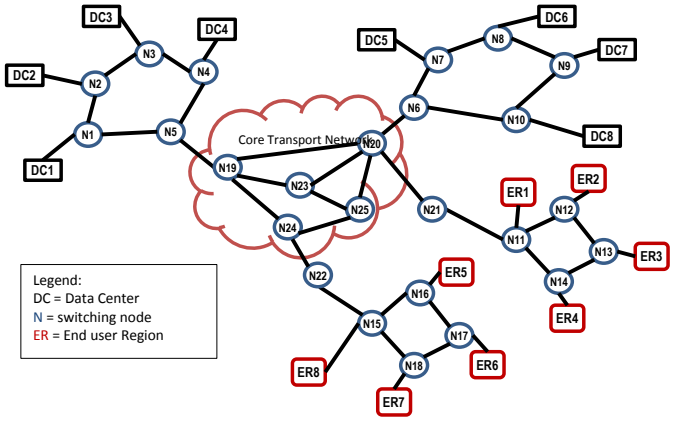


Partial VN View (H) does give <u>more</u> bottleneck Information for some key bottlenecks

Link	Capacity (Gbps)	Cost
N5-C1	100	1
N8-C1	50	1
N21-C1	50	1
N22-C1	25	1
N1-N5		
.....		
ER8-N17	10	1

Figure A.4 – Partial graph topology view, high granularity

Figure A.4 is one step more granular than Figure A.3. The core transport network is represented as one virtual node C1 in this figure. This level of VN abstraction gives a more granular view of topology and a more detailed level of capacity information.



- **Full Network View provides full details of all link information.**
- **Link cost can be represented as latency, max b/w available, link utilization, distance or any combinations of multiple costs.**

Figure A.5 – Full graph topology view, full granularity

The full granularity view of a physical network is same as the physical network itself. Some applications may need a full network view with detail link and node information. This also depends on the willingness to pay for such detail information, and is subject to the network operator's policy.

Another aspect of VN abstraction is service specific abstraction. Figure A.6 illustrates this.

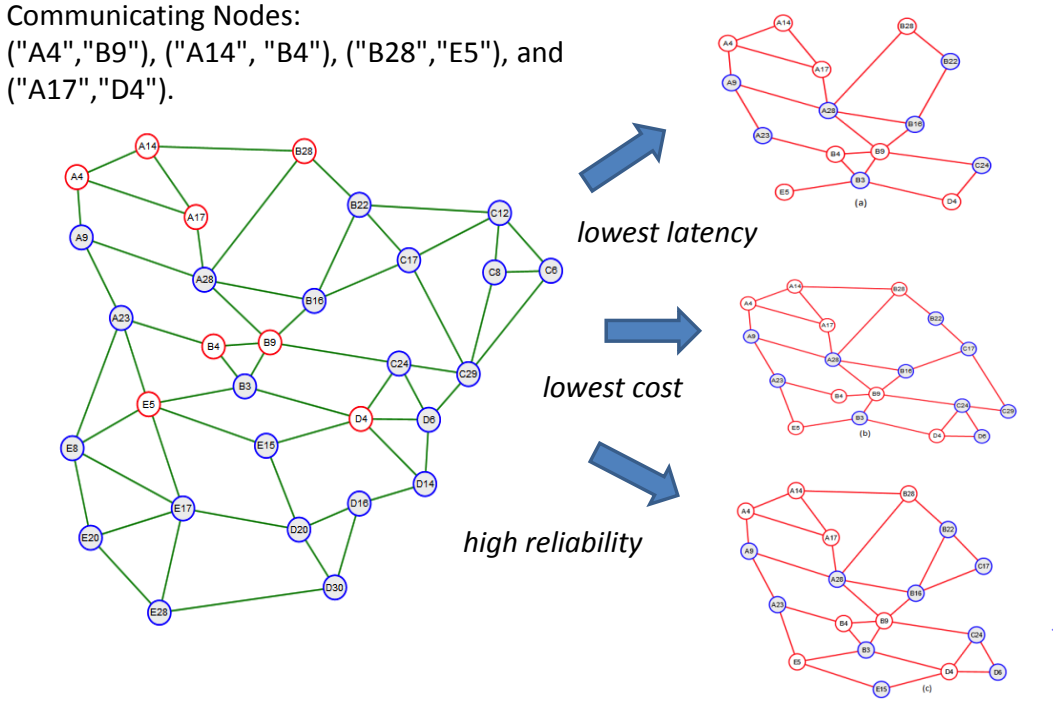


Figure A.6 – Service-specific topology reduction

The VN shown on the left side represents a full physical topology. Assume that a client application only needs four pairs of communication nodes: (A4, B9), (A14, B4), (B28, E5), and (A17, D4). The figures on the right show different topologies generated by a k-shortest path algorithm for different objectives of optimization: lowest latency, lowest cost, and highest reliability, respectively. The resulting graphs show different VN abstractions depending on the objective function. They also prune irrelevant nodes and links from the graph. This illustration shows the need for objective functions, constraints, information hiding, and reduction to be able to represent client service-specific topology abstraction.

In summary, there are a number of key factors in VN formulation, providing the means to express different levels of abstractions.

- Granularity of VN:
  - Endpoint-only view (lowest granularity, similar to Figure A.2)
  - Graph view with varying levels of granularity
- Objective function of topology
  - General (no particular objective function is associated with the topology)
  - Service-specific (latency minimal path, reliable path, maximum reservable bandwidth path, etc.)
  - Multi-objectives (a combination of multiple objectives)
- Information hiding and reduction

- This is subject to the policy of the service provider and negotiation between client and service provider.
- Information details may depend on the pricing model. The willingness to pay more for details may be considered in the service provider's pricing model. (Note that a higher degree of information detail exposes more dynamic aspects to be handled by the DC controller. This may involve, for example, topology changes that do not affect existing or potential services, but are exposed by the VN.)

Appendix B: Acronyms

AAA	Authentication, authorization, and accounting
AIS	Alarm indication signal
BER	Bit error rate
BFD	Bidirectional forwarding detection
CCM	Continuity check message
CDPI	Control-data plane interface
CFM	Connectivity fault management
CVNI	Control-virtual network interface
CWDM	Coarse wavelength division multiplexing
DC	Data center
DCC	Data center controller
DCI	Data center interconnection
DWDM	Dense wavelength division multiplexing
EBR	Excess burst rate
EMS	Element management system
EPID	Endpoint identifier
FEC	Forward error correction
GbE	Gigabit Ethernet
LAG	Link aggregation group
LLDP	Link layer discovery protocol
LOS	Loss of signal
MAC	Media access control
MEF	Metro Ethernet Forum
MEP	Maintenance association end point
MIP	Maintenance domain intermediate point
MPLS	Multi-protocol label switching
NE	Network element
NMS	Network management system
NNI	Network-network interface
NPC	Network provider controller
OEO	Optical-electrical-optical conversion
ONC	Optical network controller
OTN	Optical transport network
PCE	Path computation element
PM	Performance monitoring
POI	Packet-optical integration
PoP	Point of presence
PW	Pseudo-wire
QoS	Quality of service
ROADM	Reconfigurable optical add drop multiplexer
ROM	Read-only memory
SDN	Software-defined networking
SLA	Service level agreement
SNR	Signal to noise ratio

SRG	Shared risk group
TDM	Time division multiplexing
UNI	User-network interface
VID	VLAN identifier
VLAN	Virtual local area network
VM	Virtual machine
VN	Virtual network
VNE	Virtual network element
VNS	Virtual network service
WAN	Wide area network
WDM	Wavelength division multiplexing

Appendix C: Glossary

Note: There is active work on terminology in the Optical Transport Working Group and across ONF. The terms used in this document are defined provisionally to facilitate discussion. It must be understood that their definitions may be slightly or completely revised when final agreement is achieved.

Control data plane interface (CDPI). The reference point for all protocols between an SDN controller and an instance of its agent in the data plane.

Control virtual network interface (CVNI). The interface for service advertisement and activation between the DC controller and NPC.

Data center (DC) operator. An entity responsible for providing high capacity compute and storage services to clients, and in this use case a client of the service provider.

Data center controller. A client of the provider network controller. It is a software agent operating on behalf of the DC operator and is responsible for coordinating WAN resources to meet the requirements of the applications hosted in the DCs.

Optical network controller. A functional component in some decompositions of an SDN controller, responsible for controlling the transport network.

Provider network. A transport network whose resources are available to SDN clients.

Network provider controller (NPC). A software agent operating on behalf of the service provider. It is responsible for advertising to clients (e.g., the DC controller) available connectivity services and instantiating services requested by those clients.

Service provider. The entity responsible for providing WAN services to clients. It may also be the same as the network provider when it owns the entire transport network providing the network services.

Transport network. A network that transparently delivers traffic among geographically separated endpoints. Networks based on L0, L1, L2 technology are typically considered to be transport networks.

Virtual network. The subnetwork of a provider network that represents the environment of a given client.

Contributors

Malcolm Betts
Tae Sang Choi
Nigel Davis
Paul Doolan
Luyuan Fang
Trevor Graham
Marshall Ha
Jia He
Dave Hood (Editor, Use Case 2)
Richard King
Kam Lam
Young Lee (Editor, Use Case 3)
Victor Liu
Victor Lopez
Ben Mack-Crane
Geoffrey Mattson
Piotr Myslinski (Editor, Use Case 1)
Lyndon Ong (Editor, Use Case 4)
Eun Kyoung Paik
Ping Pan
Jonathan Sadler
Mohammad Sarwar
Karthik Sethuraman
Sejun Song
Eve Varma
Maarten Vissers
Won Dong Yun
Xueqin (David) Wei

Open Networking Foundation / www.opennetworking.org

The Open Networking Foundation is a nonprofit organization founded in 2011, whose goal is to accelerate the adoption of open SDN. ONF emphasizes the interests of end-users throughout the Data Center, Enterprise, and Carrier network environments.

[1] Metro Ethernet Forum, MEF 6.1 (2008), Ethernet services definitions, phase 2
[2] Metro Ethernet Forum, MEF 10.2 (2009), Ethernet services attributes, phase 2
[3] IEEE 802.1X (2010), Port-based network access control
[4] IEEE 802.1ag (2007), amendment 5 to 802.1Q-2005, Connectivity fault management

[5] IEEE 802.1AB (2005), Station and media access control connectivity discovery
[6] Ibid.