OPTICAL INTERNETWORKING FORUM

OPEN NETWORKING FOUNDATION

SDN Transport API Interoperability Demonstration

OIF/ONF Whitepaper
February 10, 2017
www.oiforum.com
www.opennetworking.org

Contents

1 Executive Summary 3
2 Introduction 5
3 Demonstration Set-up 6
3.1 Worldwide Test Topology 6
3.2 Testing Methodology 7
4 T-API Modules Tested 7
4.1 Topology Service 8
4.2 Connectivity Service 9
4.3 Notification Service 10
5 Use Cases Tested...11
5.1 Use cases in the context of ETSI-NFV architecture...........................11
5.2 Multi-Domain Service Provisioning 12
5.3 Low Latency L0 Path across Metro/Regional Data Centers 15
5.4 Variable Bandwidth Paths Across a Core 16
6 Findings 18
7 Benefits..19
7.1 Benefits of Interop Testing Methodology 19
7.2 T-API Benefits 20
8 Conclusion 20
1 Appendix A: List of Contributors..21
2 Appendix B: About the OIF 21
3 Appendix C: About the ONF..21
4 Appendix D: Glossary 22
5 Appendix E: The Transport API Standard 22

1 Executive Summary

The Transport SDN market is driven by the need to accelerate service provisioning to meet dynamic and on-demand characteristics of cloud and edge applications. Market adoption of transport SDN is steady and determined. Carriers are planning, trialing and deploying transport SDN, mostly in contained, greenfield domains. With accelerated service provisioning as the driving application, market focus is aligning on network automation and programmability capabilities and the expected operational simplification benefits (versus Capex savings). In turn, this is driving a need for open systems/software, flexible and disaggregated functional components as well as flexible transport mechanisms and programmable wavelength modulation schemes.

The trigger point for significant commercial deployments (i.e. mass adoption) of transport SDN may depend on defining, testing and assuring interoperability of key network functions and interfaces. To that end, the OIF developed and published a Transport SDN Framework that defines key functions and interfaces. In 2014, the OIF partnered with the Open Networking Foundation (ONF) to conduct an interop demo in 2014 that tested pre-standard ONF OpenFlow extensions for the Southbound Interfaces or Application Programming Interfaces (APIs) and prototype transport Northbound APIs to support Service and Topology requests. That work led to the initiation of standards work in ONF on the Northbound Transport API (T-API) and approval of T-API specs in 2H2016.

In the 2016 OIF SDN Transport API Interoperability Demonstration the OIF and ONF partnered to lead the industry toward the wide scale deployment of commercial SDN by testing ONF T-API standards. The interoperability test and demonstration, managed by the OIF, addressed multi-layer and multi-domain environments in global carrier labs located in Asia, Europe and North America.

The definition of T-API standards within the ONF is a pragmatic approach to obtain an end-to-end (E2E) Software Defined Network (SDN) infrastructure for carrier networks. It allows network programmability to be implemented across domains without requiring full interoperability between each network element. Deployment of T-API as a North-Bound Interface (NBI) of optical controllers allows the utilization of a common abstraction model to support optical services.

Interop testing in carrier labs under the OIF umbrella allows carriers to have direct visibility into T-API implementations to assure interoperability of different vendors across the T-API interface. Additional use cases based upon the API standards are clarified in the testing and may be defined through OIF implementation agreements to provide a common set of requirements. Participants in the OIF Transport SDN interoperability event also submitted a proof of concept demo proposal to ETSI NFV. The proposal, "Mapping ETSI-NFV onto Multi-Vendor, Multi-Domain Transport SDN", was accepted.

The testing successfully demonstrated that T-API enables real-time orchestration of on-demand connectivity setup, control and monitoring across diverse multi-layer, multi-vendor, multi-carrier networks. Some functional as well protocol related issues and gaps were identified with the transport APIs. The experiences of the testing will be shared across the industry to help develop critical implementation agreements and specifications. The goal is to accelerate service provisioning to meet dynamic and on-demand characteristics of cloud and edge applications.

2 Introduction

Operators today are looking towards Software Defined Networks (SDN) to enable programmability of their networks for efficiency, speed of deployment and new revenue-generating network services. Widespread adoption of the programmability paradigm depends on the availability of common or standardized APIs that allow access to domain specific attributes and mechanisms without requiring the API itself to be specific to the vendor or technology. The Transport API (T-API) is designed to allow network operators to deploy SDN across a multi-domain, multi-vendor transport infrastructure, extending programmability across their networks end-to-end.

By abstracting the details of the lower level Domain, T-API supports integration of Domains of different technology and different vendor equipment into a single virtualized network infrastructure:

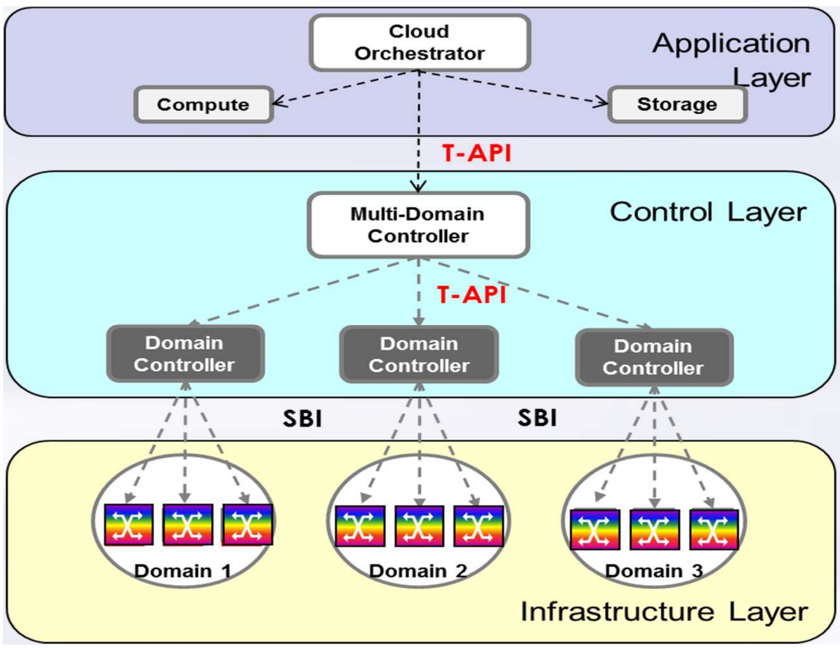


Figure 1: T-API Multi-Domain, Multi-Vendor Integration

3 Demonstration Set-up

3.1 Worldwide Test Topology

Testing was carried out in 5 carrier labs in Asia, Europe and North America as shown below in Figure 2:

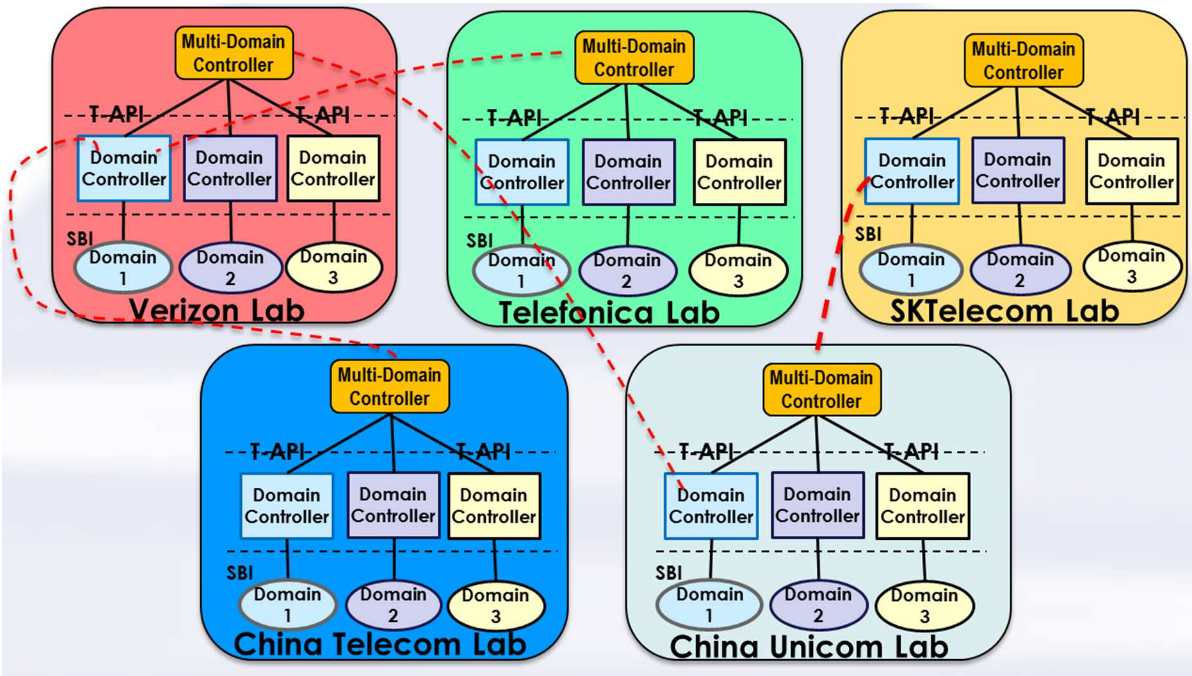


Figure 2: Worldwide Test Topology

Participating Carriers:

- Asia: China Telecom, China Unicom, SKTelecom
- Europe: Telefonica
- North America: Verizon

Eleven vendors and two research institutes participated with software or hardware, providing a variety of types of equipment and software functions. Two consulting carriers also provided support and monitored results. The participating vendors included:

- ADVA Optical Networking
- Ciena
- Coriant
- FiberHome
- Huawei Technologies Co., Ltd.
- Infinera
- Juniper Networks
- NEC Corporation
- Sedona Systems
- SM Optics
- ZTE

Research Institutions:
- China Academy of Telecommunications Research
- Centre Tecnològic Telecomunicacions Catalunya

Consulting Carriers:
- TELUS
- Orange

3.2 Testing Methodology

One of the key characteristics of the demonstration was the ability to test applications, controller implementations and optical network elements implemented by different organizations, interoperating through prototype standard or common interfaces.

In the preparation phase participants cooperated in defining common test specifications defining the usage of T-API elements and protocol details and common test case specifications defining the set of procedures and sequence of protocol messages to be exchanged between systems.

In the intra-lab phase testing was conducted between Multi-Domain Controller implementations and Domain Controller implementations within individual carrier labs, including verification of data plane connectivity after setup of a connectivity service. In addition to testing of the full complement of API requests and responses, several use cases were tested as discussed below to demonstrate real world applications of T-API.

Finally, in the inter-lab phase, testing was conducted between Multi-Domain Controllers and Domain Controllers in different carrier labs to allow for additional matches between participants and observation by the participating carriers of implementations in remote labs. Data plane connectivity between labs was simulated rather than true physical connections due to cost and complexity.

4 T-API Modules Tested

The ONF Transport API (T-API) functional requirements are defined in the ONF TR-527 (Functional Requirements for Transport API, June 2016). For the initial version of T-API this includes 5 functional modules supporting different services:
- Topology Service
- Connectivity Service
- Path Computation Service
- Virtual Network Service
- Notification Service

Figure 3 presents top level decomposition of the T-API defined by [TR-527]. For the 2016 Demo 3 main modules were supported: Topology, Connectivity and Notification.

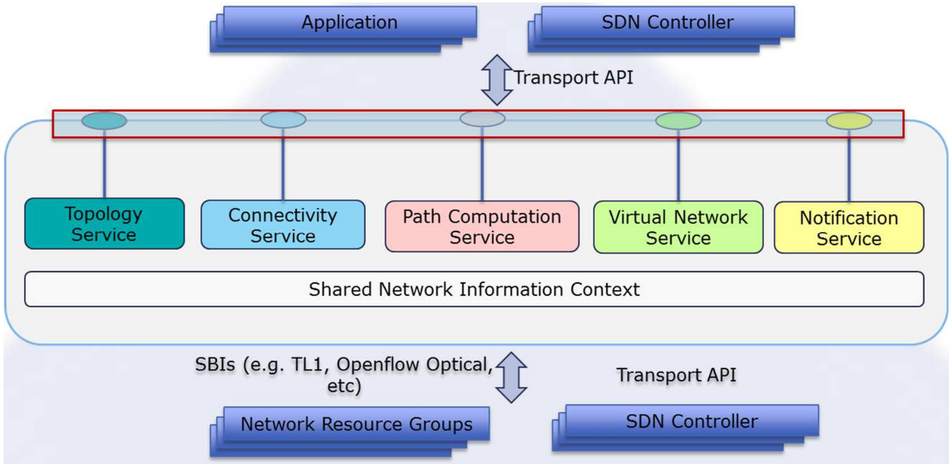


Figure 3: Top level decomposition of the T-API Services [1].

The following sections discuss the specific module functions used in testing. More information about the T-API work at ONF is contained in Appendix E to this document.

4.1 Topology Service

Topology Service is first out of five T-API services as defined by ONF [TR-527]. It allows to retrieve the most basic data about controlled networks – abstract view of devices and the way how they interconnect. As a basic one, apart from providing important network inventory information, it gives a foundation to:

- other parts of T-API - Path Computation, Connectivity and Virtual Network,
- potential applications, which define services on top of collected topological data (like monitoring, planning, optimization and others),

Figure 4 presents key objects in the Topology module with their attributes and dependencies [T-API SDK, http://github.com/OpenNetworkingFoundation/Snowmass-ONFOpenTransport/]. For the simplicity reasons attributes of composite types (referred also as “topology-pacs”) were hidden.

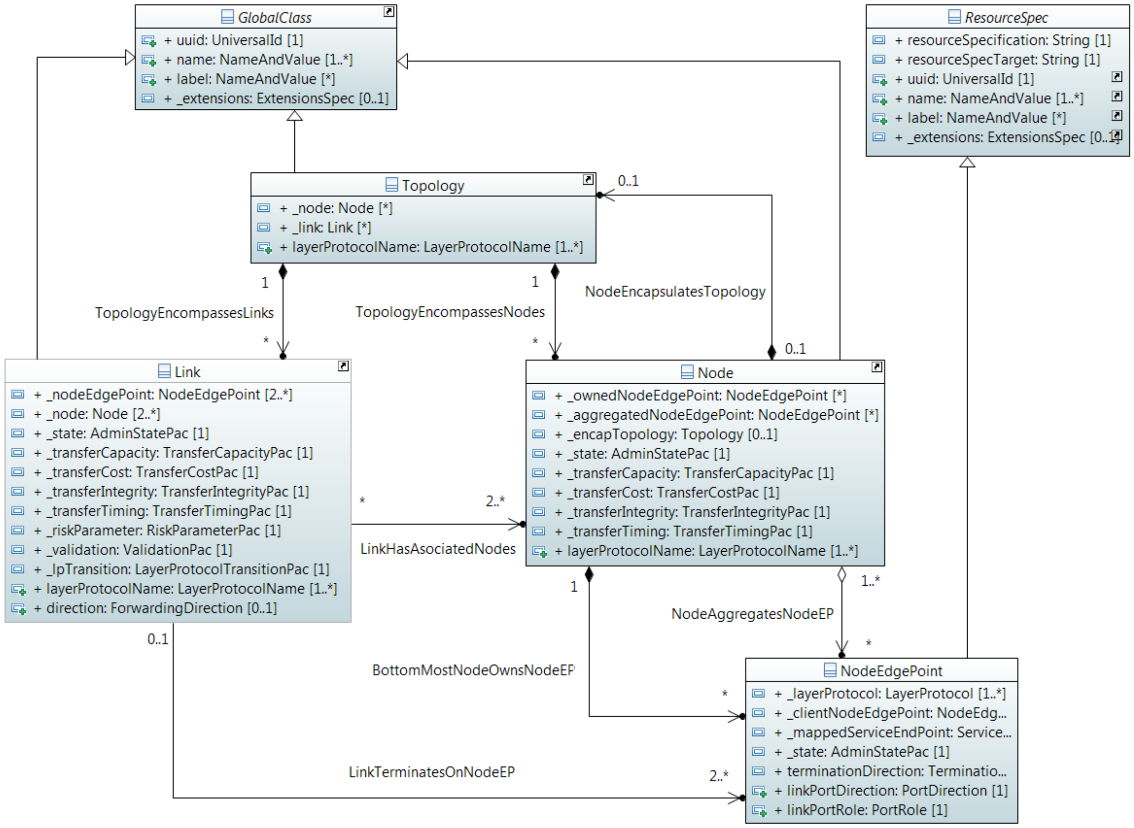


Figure 4: Key objects of Topology module.

The Topology module enables Retrieval for the objects in a Context shared between provider and client. Typical functional operations are defined as follows:

- get list of topologies - a collection of references to Topology objects contained in the shared context,
- get details of a given topology - attributes of the Topology object, collection of references to Node and Link objects contained in a specific Topology,
- get list of nodes – a collection of Node objects in a given Topology,
- get details of a given node – attributes of the Node object,
- get list of links - a collection of references to Link objects contained in specific Topology,
- get details of a given link – attributes of the Link object,
- get list of node edge points - a collection of references to NodeEdgePoint objects contained in specific Topology and Node
- get details of a given node edge point – attributes of the NodeEdgePoint object

4.2 Connectivity Service

The Connectivity Service supports operations related to the lifecycle of a connectivity service between two or more endpoints at the edge of a Transport network. To provide this support, it enables the Creation, List, Query, Retrieval and Deletion of a connectivity service as well as associated objects (i.e. connection and serviceEndpoint). Connectivity Services are defined by the following basic attributes:

- a set of ServicePorts that will be connected, identified by the associated Service EndPoints, their roles in the service and the ServiceLayer to be used at the Service EndPoint
- a set of ConnectivityConstraints that define the type of service (e.g., point-to-point), the requestedCapacity and a variety of potential path constraints such as:
    - latency requirement
    - cost requirement
    - requirements for inclusion or exclusion of topology elements such as nodes or links
    - requirements for diversity or co-routing with existing connections

The T-API server responds to the request for Connectivity Service with a unique identifier for the service, the lifecycle state of the service, plus details of the service such as the constraints that have been met and optionally the identifiers and details of supporting transport connections.

The initial response from the server may be returned before the Connectivity Service is fully implemented, for example, in cases where photonic connections are needed and some latency for tuning and balancing the photonic elements is incurred. In this case the initial lifecycle state indicates that the service is still in Potential rather than Installed state. Determination of when the service transitions to Installed state can be done either by polling for the service's lifecycle state using the retrieval functions discussed below, or by using Notification service to request notification of state changes in the service.

Connectivity Service clients can also retrieve information about connectivity services such as:

- a list of the identifiers for all active connectivity services
- detailed attributes for a particular connectivity service, based on its unique identifier
    - including its current lifecycle state (this can be used to poll for when the service enters Installed state as discussed above)
- a list of the identifiers for connections supporting a connectivity service
- detailed attributes for a particular connection, based on its identifier

For the Interop demo, only Ethernet point-to-point private line services were tested as this was supported by all vendors and their domains. Internally, the Ethernet service was transported over packet, OTN ODU or OTN OCh switched networks depending on the particular vendor and domain.

4.3 Notification Service

The Notification Service supports autonomous notification from the network of significant events, such as failure of an element of the network topology or change of state of a connectivity service. The Notification Service supports the following functions:

- Discovery of the supported notification types – these include notification of object creation and deletion as well as state changes or attribute value change
- Creation, modification and deletion of a notification subscription – allows the client to subscribe to notifications via websocket of events, and to modify and delete this subscription
- Suspend and resume notification – allows the client to temporarily suspend receiving notifications and then later resume them

In the demonstration, a controller that did not support notification would need to be periodically polled for any change in state or topology. A controller supporting notification would provide an address that the client would connect to via websocket to receive any subsequent notifications for which they had created a subscription.

5 Use Cases Tested

5.1 Use cases in the context of ETSI-NFV architecture

Since the transport networks generally reside at the lower layers of the networking infrastructure hierarchy, the Transport SDN use cases are often more relevant when considered in the context of a larger service and user ecosystem. For this interoperability demonstration, the use cases tested were framed in the context of the ETSI-NFV architecture as depicted below.

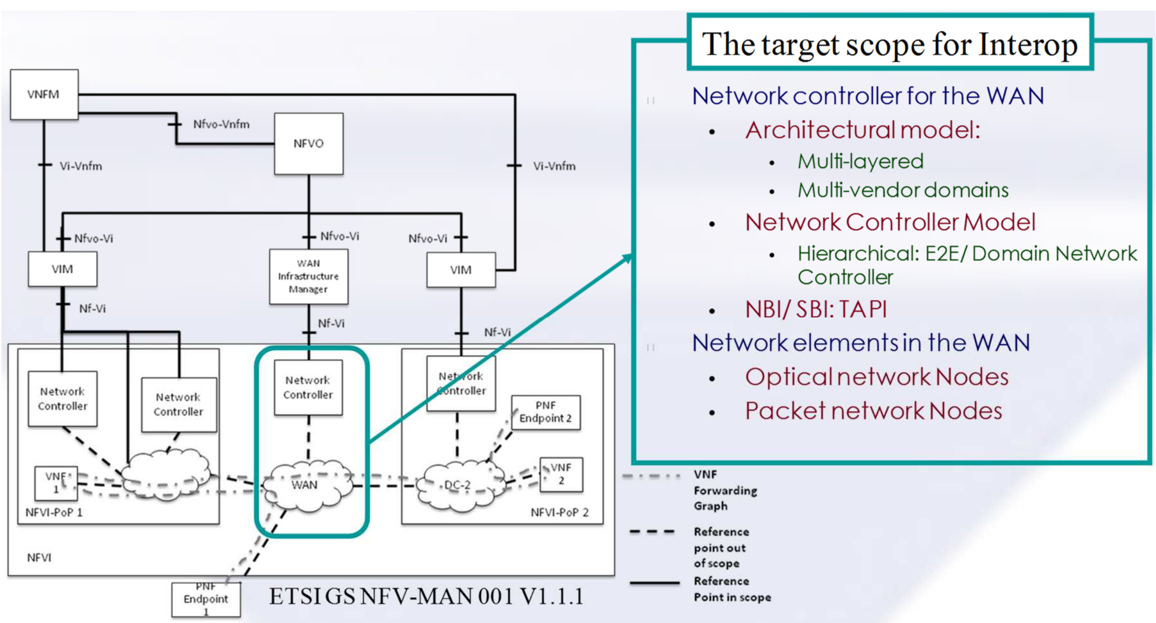


Figure 5: ETSI NFV Framework

ETSI-NFV use cases describe multiple sites hosting NFVI-POPs, which are interconnected over a Wide Area Network (WAN) infrastructure. As shown in Figure 5, the network is architecturally configured by a network controller, interfacing with WIM, and Wide Area Network (WAN) infrastructure. The WAN interconnects multiple ETSI-NFV sites. The use cases in this interoperability test aim to demonstrate connectivity life cycle management using SDN Network Controllers for geographically distributed ETSI-NFV site interconnections.

In the test setup, the network infrastructure is assumed to consist of multiple network domains of individual vendors and operators. A domain controller is responsible for a domain network infrastructure. A multi-domain controller is responsible for end-to-end connectivity of the network infrastructure. The multi-domain controller implements the T-API interface to WIM as well as the individual domain controllers.

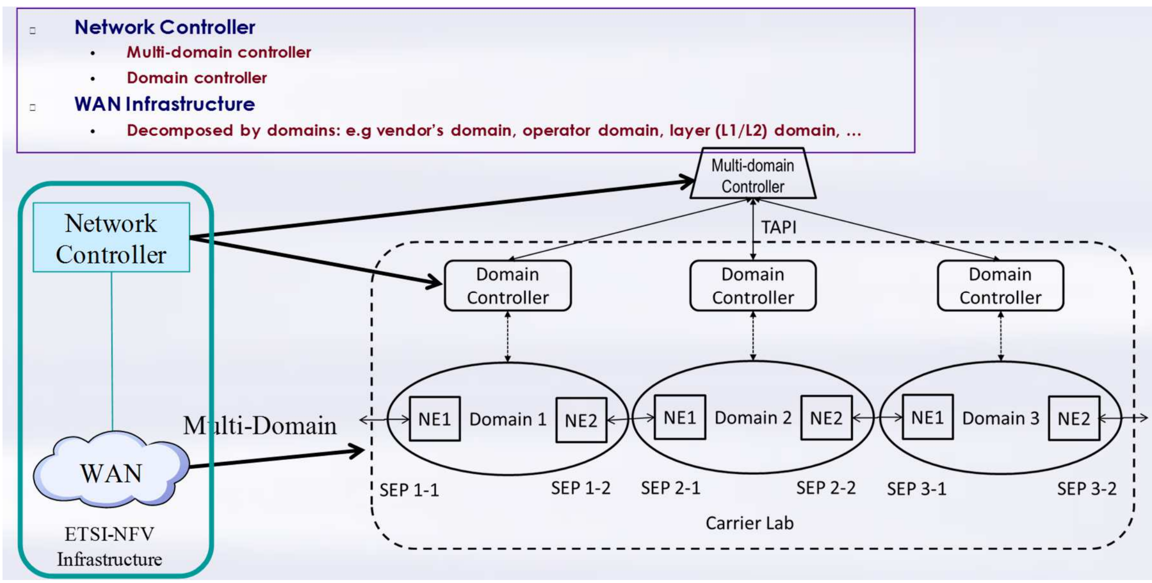


Figure 6: Mapping to T-API Testing

Participants in the OIF Transport SDN interoperability event also submitted a proof of concept demo proposal to ETSI NFV. The proposal, "Mapping ETSI-NFV onto Multi-Vendor, Multi-Domain Transport SDN", was accepted and details can found in the following ETSI-NFV wiki page. The open demonstration of NFV concepts in a Proof of Concept (PoC) helps to build industrial awareness and confidence in NFV as a viable technology. Proofs of Concept also help to develop a diverse, open, NFV ecosystem. Results from PoCs may guide the work in the NFV ISG by providing feedback on interoperability and other technical challenges.

5.2 Multi-Domain Service Provisioning

One of the main Use Cases tested was Multi-Domain Connectivity Service with local and end-to-end path constraints and local and end-to-end recovery. This

Use Case involved several steps executed by a Multi-Domain Controller working with multiple lower level Domain Controllers as shown in the following figure:

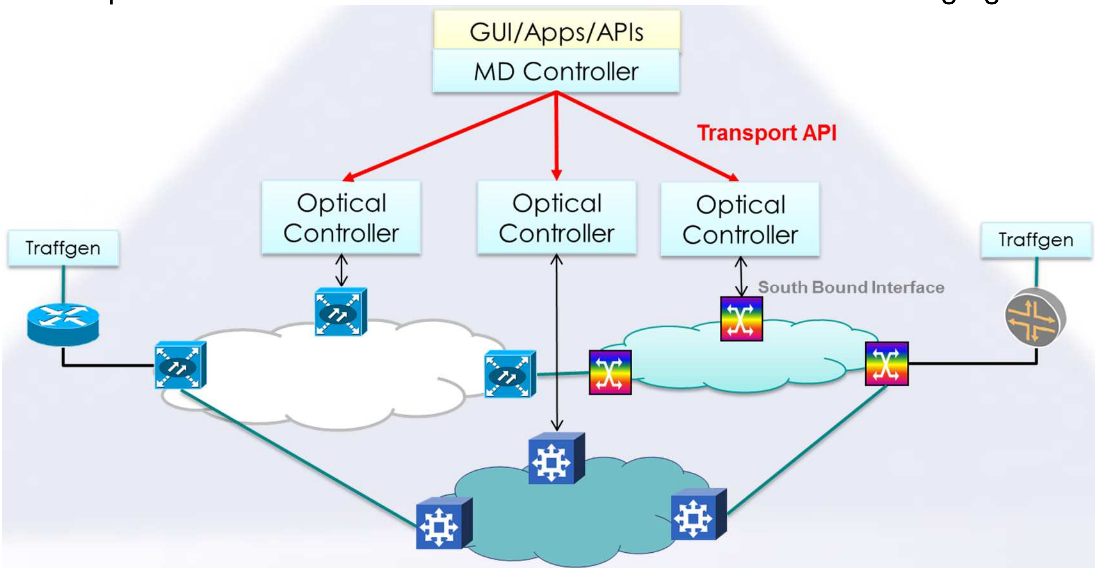


Figure 7: Multi-Domain Topology for Connectivity Service Use Case

In the initial stage of the Use Case, the MD Controller queries its Domain Controllers for their topology information using the Topology API. Based on the retrieved information and internal knowledge of the inter-domain links, the MD Controller then builds a multi-domain topology that can be used to compute paths for new services.

In the next stage of the Use Case, the MD Controller builds a multi-domain connectivity service using the Connectivity Service API to create the required services in each domain. At first, a service is built without specifying path constraints, allowing each Domain Controller to perform internal path computation per its local optimization, e.g., using shortest path. After each Domain Controller indicated that their portion of the service was installed, the data plane connectivity was tested if possible (for services spanning domains in different labs, data plane connectivity could not be tested as there was no actual capacity available between labs).

In subsequent states of the Use Case, the MD Controller built multi-domain connectivity services using local and end-to-end path constraints, e.g.,

- specifying in the Connectivity Service API that the Domain Controller should use diverse or co-routed paths or include specified links in the path of the service in order to exercise the ability of the Connectivity Service API to carry Connectivity Constraints
- using path computation in the MD Controller to determine end-to-end path characteristics such as use of a diverse domain path, and then using the Connectivity Service API with the associated Service EndPoints to the Domain Controllers along the end-to-end path

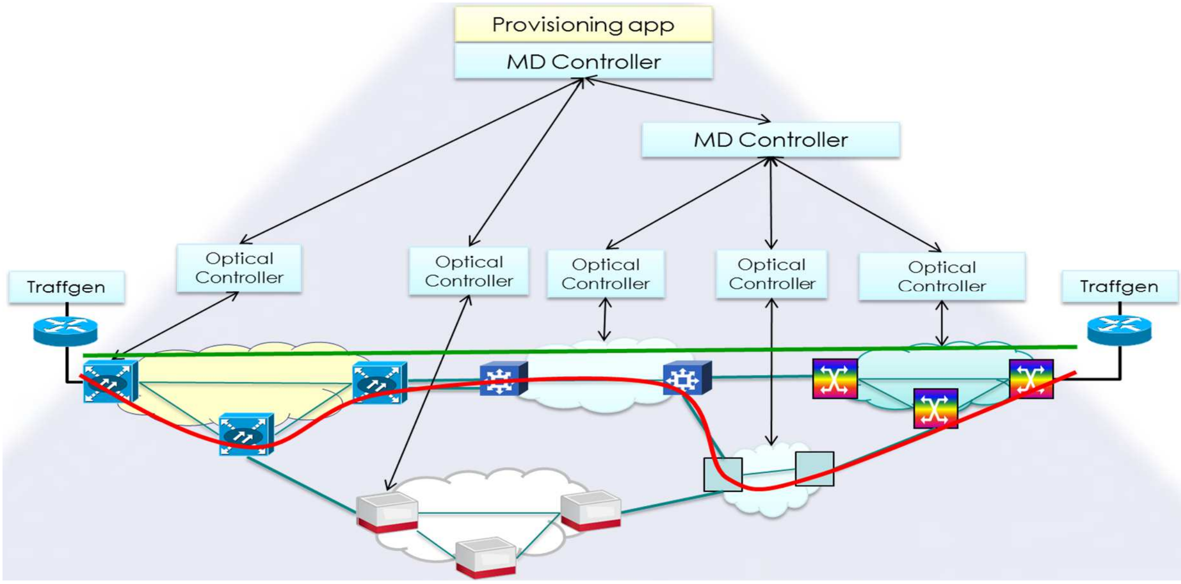


Figure 8: Connectivity Service with Local Path Diversity Constraints

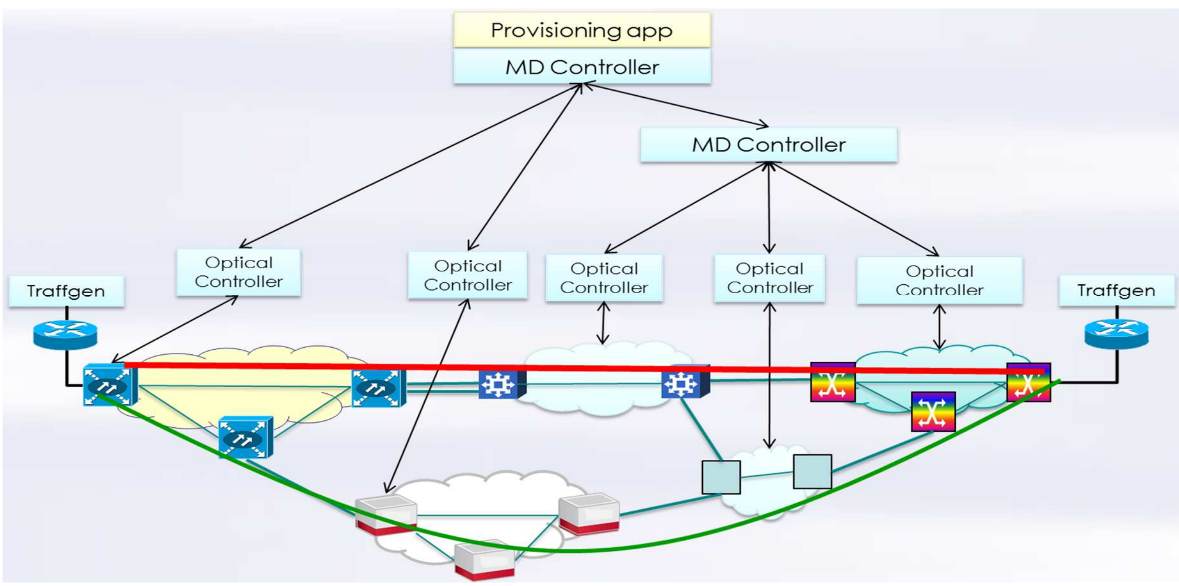


Figure 9: Connectivity Service with End-to-End Diversity Constraints

Finally, local and end-to-end recovery was specified in the Use Case, where these involved the following procedures:

- For local recovery, simulating a failure within a domain and triggering that domain's internal recovery functions so that the services is restored within that domain without disturbing other domains in the service
- For end-to-end recovery, simulating a failure that was not fixable within the associated domain and using the MD Controller to provision restoration of the service across a path bypassing the affected domain.

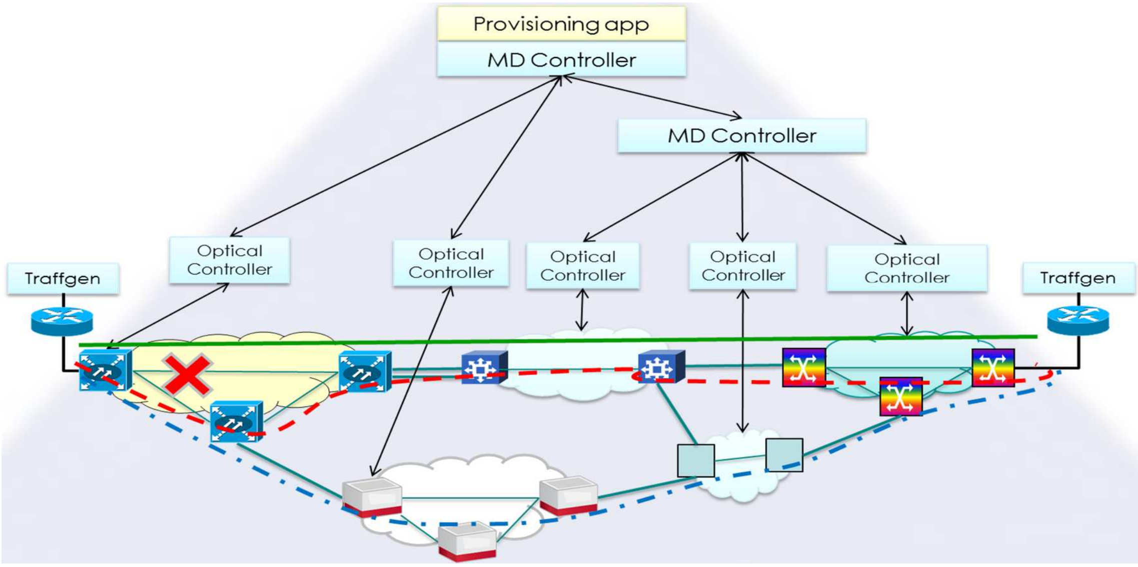


Figure 10: Local and End-to-End Recovery

5.3 Low Latency L0 Path across Metro/Regional Data Centers

Another use case tested was establishing a low-latency L0 path across representative Metro/Regional Data Centers. Using the controller hierarchy outlined in Fig. A below, the Multi-Domain Controller receives a connectivity service request for low latency connectivity between regional PoPs (N2 - N3) (See Fig 11 below).

The Multi-Domain Controller first builds the multi-layer, multi-domain topology by querying the domain controllers using the Topology API. The Multi-Domain Controller uses this topology information to compute an all optical L0 path avoiding the core network.

The Multi-Domain Controller then issues connectivity requests via the Connectivity Service API to complete install the calculated path. Video traffic traverses the path and is validated in real-time using dual video monitors.

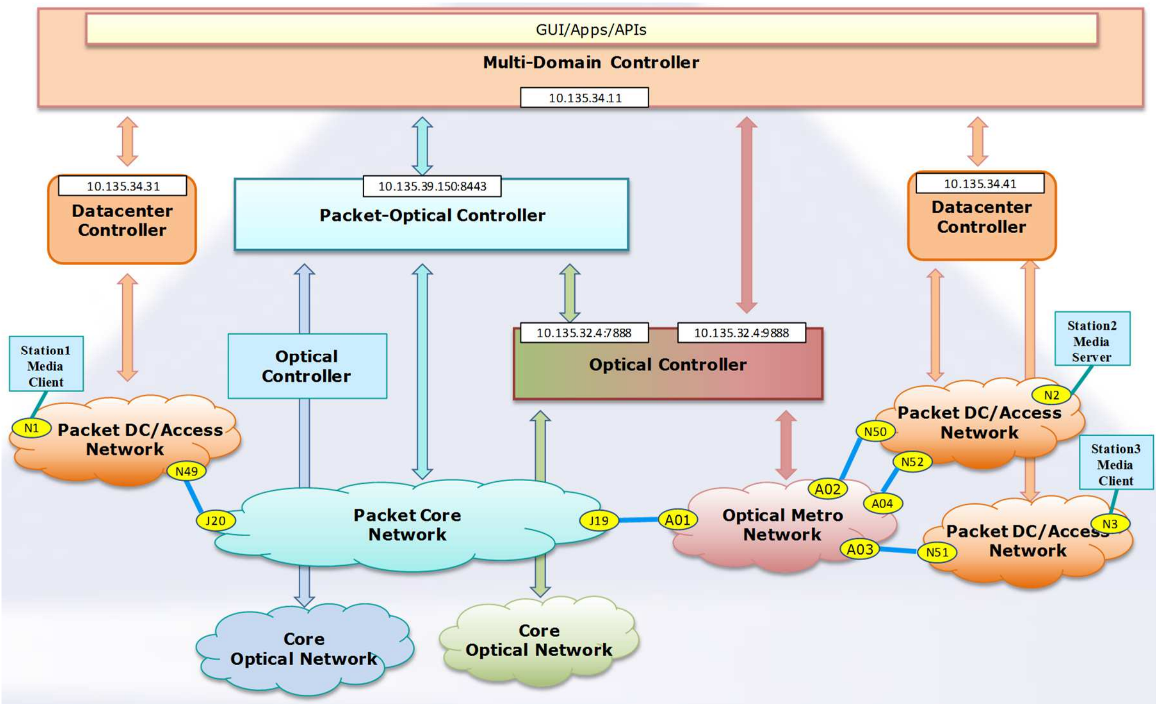


Figure 11: Transport SDN Controller Hierarchy

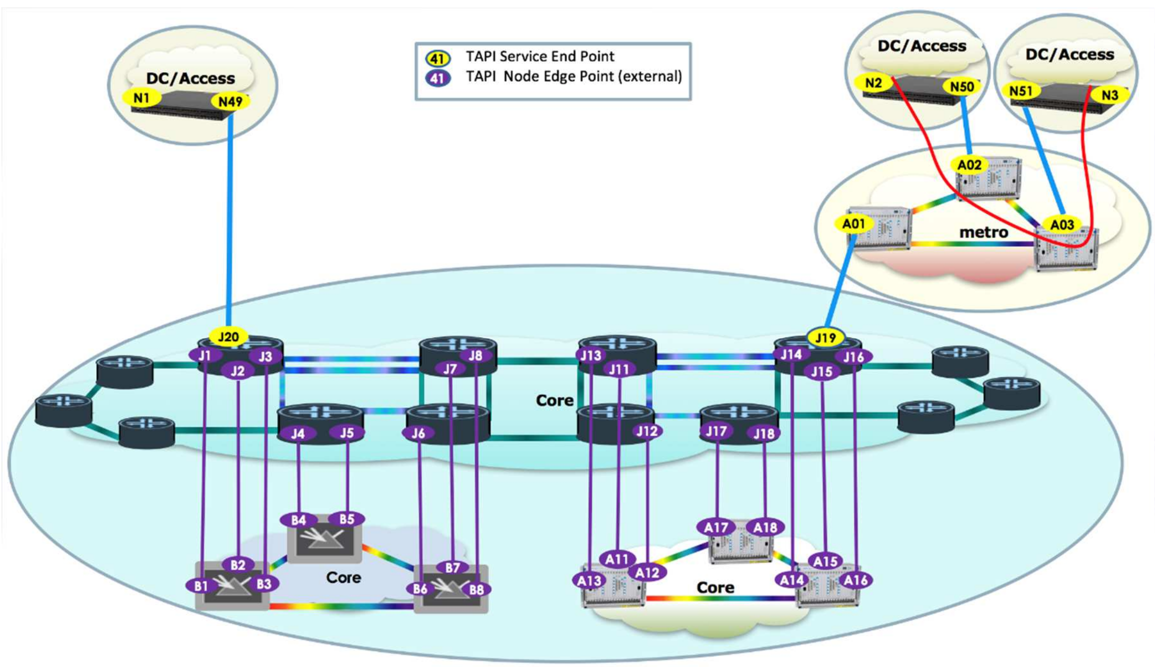


Figure 12: Low Latency L0 Path across Metro/Regional Data Centers

5.4 Variable Bandwidth Paths Across a Core

Testing of connectivity through a representative core network was accomplished using multiple vendor domains including optical and packet switched topologies as shown in Figure 13 below. The controller hierarchy is shown in Figure 10.

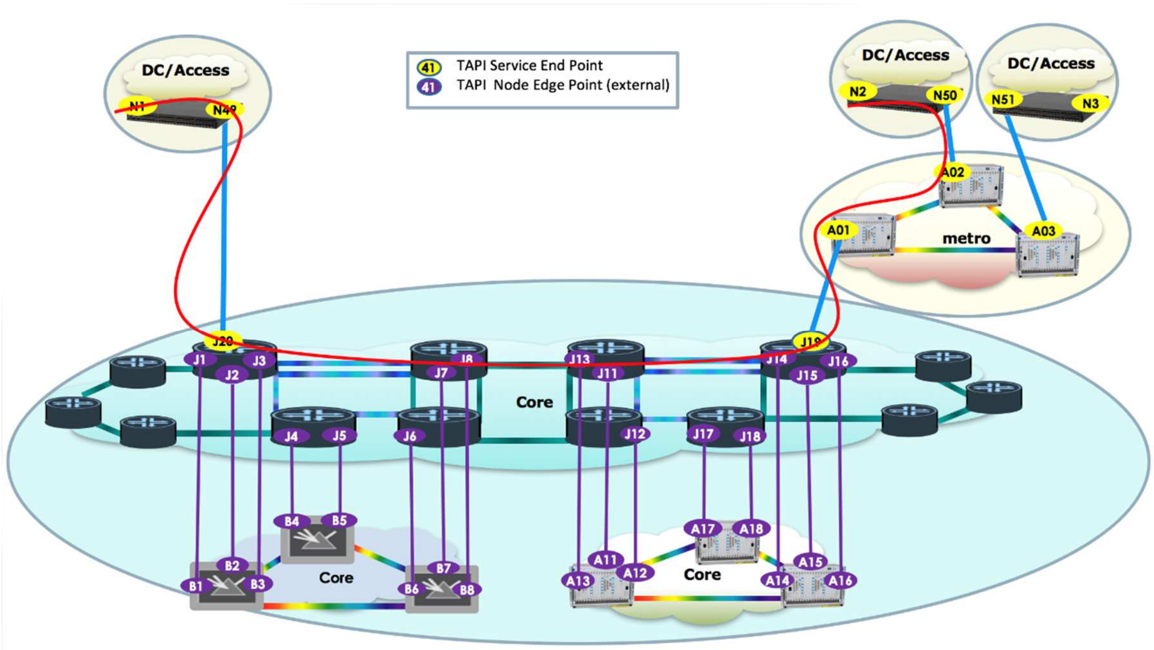


Figure 13: Variable Bandwidth Across the Core

As seen in Figure 13, the interconnected network of vendor domains represents a metro/core architecture. The first step in this test involves the Multi-Domain Controller receiving a Connectivity Service Request for a variable bandwidth connection between PoPs (N1-N2).

The Multi-Domain Controller queries the domain controllers for topology using the Topology API, and uses this information to calculate a path across the network.

The Multi-Domain Controller then issues a Connectivity Service Request to the Packet Core Multi-domain controller requesting variable connectivity between J19 and J20.

The Multi-Domain Controller then makes Connectivity Service Requests to the optical controllers to establish the connectivity through the L0 networks.

Finally, the Multi-Domain Controller requests the connectivity service between the PoPs and Optical ports (N01 - N49) and (N02 - N50), and the datapath connectivity is confirmed via video monitors.

6 Findings

SDN is not only changing the way we control and manage the network, but also is also changing the manner and speed of standardization. More emphasis is being placed on following an agile process for faster and incremental implementation and deployment of the technology. Keeping this in mind, the objective of the interop was not certification, but validation of the ONF T-API as a tool to enable programmability of the transport network.

To this end, it was successfully demonstrated that T-API enables real-time orchestration of on-demand connectivity setup, control and monitoring across diverse multi-layer, multi-vendor, multi-carrier networks. It was also confirmed that it was possible to seamlessly interoperate across diverse controller platforms, including open-source based as well as proprietary platforms, SDN as well as legacy management systems. It was shown that the solution could scale well using recursive hierarchical controller architecture, and interfacing with the same set of transport APIs and abstraction concepts at every level of controller hierarchy.

At the same time, some functional as well protocol related issues and gaps were identified with the transport APIs. Examples include:

- Differences in how different domain controllers abstract and model the network
    - Example: using Unidirectional v/s Bidirectional end-points and links
    - Resolution: would require the multi-domain controller to perform appropriate mapping and service decomposition
- Expressing connectivity constraints to sufficient detail in a consistent manner
    - Example: Muxponder Port Restrictions where ingress port number == egress port number
    - Resolution: Included “Node constraints” feature into next version of ONF T-API spec, that would enable specifying rules and constraints in a standard manner
- Division and allocation of responsibilities between control systems
    - Example: Is the detailed path computation performed by local/domain controller or the higher-level multi-domain controller
    - Resolution: PCE is a sophisticated function requiring a hybrid approach where different levels of path Computation are performed in different controllers
- Synchronizing and verifying the connectivity setup
    - Example: Variation in the latency to setup photonic connections is significantly different from vendor to vendor
    - Resolution: Require asynchronous notification or polling mechanisms even for support of connectivity service setup to determine the stable state beyond the initial response to the service request
- Differences in domain controller preferences for API styles in T-API

- o Example: Some domain controllers have preferred to use the RPC style encoding of the RESTConf specification (of the T-API YANG model) while other have preferred the SCRUD flavor
- o Resolution: The T-API was amended to define a common data model that was shared by both the RPC and the SCRUD envelopes. This made it easier for the multi-domain controller to implement both API mechanisms.

Many of the findings from the interop are already being incorporated into the next version of the ONF T-API specification. It should be acknowledged that these important findings could have only been facilitated through such interop testing amongst multiple vendor and controller systems.

7 Benefits

7.1 Benefits of Interop Testing Methodology

In general, these multi-vendor interoperability tests conducted jointly with carriers in their labs provide several benefits to participating OIF and ONF members:

- Carriers influence the features and requirements of technology and get equipment that meets their needs and suggested use cases
- Interoperability of features across multiple vendors allows carriers to deploy services more rapidly
- Interoperability demonstrations help to align the proposed solutions of the vendors, while creating a competitive environment that fosters innovation and leads to new developments and product evolution
- Carriers can test vendor interoperability and equipment first hand
- Vendors lower their risk of development because of common functionality, design and component characteristics
- Vendors have a neutral ground to test implementations against others for interoperability and improve their implementations

SDN and virtualization promise to simplify optical transport network control by adding management flexibility and programmatic network element control to enable the rapid services development and provisioning. Improved network efficiency and agility will likewise deliver benefits of lower overall operational expenses and faster time-to-market/revenue resulting in improved ROI for carriers and operators. To this end, participating carriers and vendors leverage the prototype demo to gain practical experience with Transport SDN technology in real-world scenarios to assess the status of the technology, develop pertinent use cases, and identify any interoperability and operational challenges that may slow the evolution to commercial deployments. The multi-vendor nature of the testing performed in carrier labs gives carriers the confidence that different transport vendors/systems can work together.

7.2 T-API Benefits

SDN and virtualization have the promise of simplifying transport network control, adding management flexibility, and allowing the rapid development of new service offerings by enabling programmatic control of transport networks and equipment. Open well defined T-API's are required for services to become programmable. They expose the resource view and provide network functionality per service level agreements. Standards based T-API enables infrastructure agnostic service provisioning and facilitates potential to integrate carriers' green and brownfield domains into a single virtualized Transport SDN infrastructure.

With changing patterns of network usage bandwidth-on-demand services are becoming important. Regular patterns of time of day and day of week usage are seen for specific classes of users and access networks. For example; enterprise users generate most traffic during weekdays and normal business hours. On the other hand, a consumer typically generates most traffic during nights and weekends. This means that time of day sharing between enterprise and consumer usage patterns is possible. Rearranging transport network topologies to interconnect networking and computing more economically based upon time of day and day of week to better serve these predictable phases of behaviors could significantly reduce overall networking costs.

Furthermore, resiliency and resource utilization of the network can be improved through coordinated multi-layer optimization techniques implemented through a centralized network control function that includes a view of both packet and optical layer topologies and the ability to optimize resource utilization globally and steer the re-allocation of resources in response to network failures.
Feasibility of creating virtual network (or resource) slices enables the deployment of multiple logical, self-contained networks on a common infrastructure platform. With common abstraction and resource representations for uniform physical and virtual resource management and control, dynamic network resource slices can be created. Resource slicing combined with NFV will enable efficient 5G deployment.

8 Conclusion

In the 2016 OIF/ONF SDN Transport API Interoperability Demonstration the OIF and ONF partnered to lead the industry toward the wide scale deployment of commercial SDN by testing key Transport API standards. The interoperability test and demonstration, managed by the OIF, addressed multi-layer and multi-domain environments in global carrier labs located in Asia, Europe and North America. The testing successfully demonstrated that T-API enables real-time orchestration of on-demand connectivity setup, control and monitoring across diverse multi-layer, multi-vendor, multi-carrier networks. Some functional as well protocol related issues and gaps were identified with the Transport API. The experiences of the testing will be shared across the industry to help develop critical implementation agreements and specifications. The goal is to accelerate service provisioning to meet dynamic and on-demand characteristics of cloud and edge applications.

1 Appendix A: List of Contributors

Dave Brown (Editor) – Nokia

Ori Gerstel – Sedona Systems
Pawel Kaczmarek – ADVA Optical Networking
Peter Landon – Juniper
Victor López - Telefónica
Lyndon Ong – Ciena
Jonathan Sadler - Coriant
Karthik Sethuraman – NEC
Vishnu Shukla – Verizon
Ricard Vilalta – CTTC

2 Appendix B: About the OIF

Launched in 1998, the OIF is the first industry group to unite representatives from data and optical networking disciplines, including many of the world's leading carriers, component manufacturers and system vendors. The OIF promotes the development and deployment of interoperable networking solutions and services through the creation of Implementation Agreements (IAs) for optical, interconnect, network processing, component and networking systems technologies. The OIF actively supports and extends the work of standards bodies and industry forums with the goal of promoting worldwide compatibility of optical internetworking products. Information on the OIF can be found at http://www.oiforum.com.

3 Appendix C: About the ONF

Launched in 2011 by Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo!, the Open Networking Foundation (ONF) is a growing nonprofit organization with more than 140 members whose mission is to accelerate the adoption of open SDN. ONF promotes open SDN and OpenFlow technologies and standards while fostering a vibrant market of products, services, applications, customers, and users. For further details visit the ONF website at: http://www.opennetworking.org.

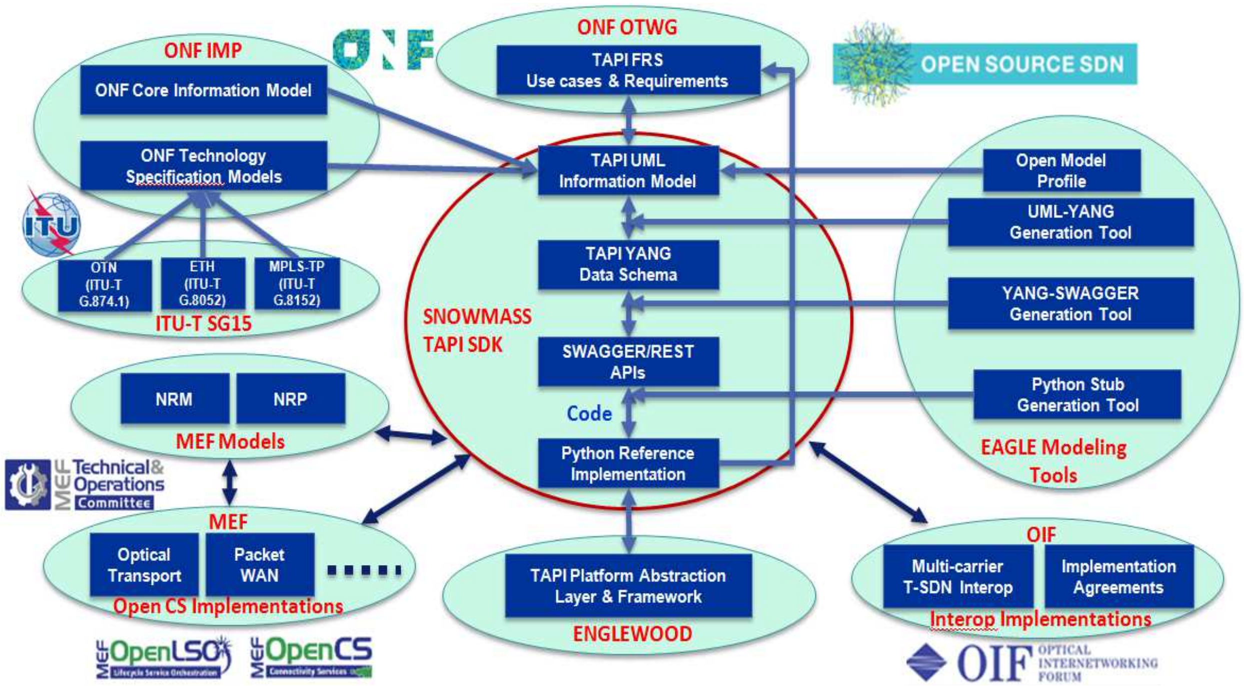
4 Appendix D: Glossary

API	Application Programming Interface
CDPI	Control to Data Plane Interface
CIM	Common Information Model
COTS	Common-off-the-shelf
CVNI	Control Virtual Network Interface
DCN	Data Communication Networks
E-NNI	External Network-Network Interface
ETSI	European Telecommunications Standards Institute
GFP	Generic Framing Procedure
IA	Implementation Agreement
ITU-T	Telecommunication Standardization Sector of the International
JSON	JavaScript Object Notation
MEF	Metro Ethernet Forum
MD	Multi-Domain
NBI	Northbound Interface
OCh	Optical Channel
ODU	Optical channel Data Unit
OF	OpenFlow
OIF	Optical Internetworking Forum
ONF	Open Networking Foundation
OpEx	Operational Expenditure
OSSDN	Open Source SDN
OTN	Optical Transport Networking
OTU	Optical channel Transport Unit
PoP	Point of Presence
QoE	Quality of Experience

5 Appendix E: The Transport API Standard

T-API is designed to be the interface between controllers at different levels of an SDN controller hierarchy, offering control over network resources at different levels of abstraction. A typical deployment would be as the interface between the Domain Controllers for several network domains and a higher level Multi-domain SDN Controller that acts as a parent. T-API has also been suggested as an interface between SDN applications (e.g., NFV MANO) and SDN controllers.

T-API is a product of the ONF Open Transport Working Group (OTWG) with input from the OIF and joint interoperability testing. T-API is closely based on the ONF's Common Information Model [CIM] developed by the ONF Information Modeling project. The CIM provides a common, technology-independent representation of data plane resources for management-control by the operator that is derived from industry models from TMF and ITU-T. The CIM effort has in



turn been adopted by organizations like TMF, ITU-T, OIF and MEF as the basis for definition work.

T-API derives its Information Model by pruning and refactoring the CIM Core Information Model as a purpose-specific realization for Transport Networks. In progressing from model to realization, T-API also incorporates work from multiple ONF Open Source SDN (OSSDN) projects:

Fig. 14: T-API components and related projects

The development of T-API has followed an agile process as described in Fig. 2. From uses cases and requirements the CIM has been pruned and refactored in a T-API UML information model, which has been used as an input to automatic translation tools to obtain T-API YANG schemas. These have been automatically translated into Swagger API descriptions that have allowed the development of a T-API reference implementation in OSSDN Snowmass.

Key features of T-API include:

- Technology-agnostic API Framework
  - Standardizes a single core technology-agnostic specification that abstracts common transport network functions for the interface
- Modular & Extensible
  - Functional features are packaged into small self-contained largely-independent modules that can be extended with technology enhancements
- SDK components generated using tools for agile prototyping
  - YANG schema generated from UML using automated tooling developed in the OSSDN EAGLE project

    - Swagger/JSON APIs generated from YANG using automated tooling developed by EAGLE following RESTCONF specifications
- Industry-wide Interoperability Objective

The ONF T-API Functional Requirements Document is publically available from ONF as TR-527, “Functional Requirements for Transport API (June 2016).

The T-API SDK is available through the OSSDN SNOWMASS project under Apache 2 license.

Open Source implementation of T-API is being pursued through the OSSDN ENGLEWOOD project.