SDN Transport API Interoperability Demonstration

OIF Whitepaper
August 7, 2018
www.oiforum.com

Table of Contents

Table of Figures

1 Executive Summary

Network operators are rapidly moving toward giving customers and their applications the ability to dynamically control services and do it in real-time. The days of waiting for service changes will soon be a thing of the past. To achieve this, they need the ability to dynamically move capacity quickly in open networks to avoid network congestion and provide better services to customers.

With network operators leading the charge for more dynamic and open networks, there must be widespread adoption of transport Software-defined Networking (SDN). By working through the specifications, rigorous interoperability testing and validation, the Optical Internetworking Forum (OIF) 2018 interoperability demo intended to substantiate Transport-Application Programming Interface (T-API) as the Northbound Interface (NBI) of choice.

As background, in 2013 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. That work led to the enhanced T-API 2.0 spec published in 2017.

In 2018 the OIF, in collaboration with MEF, introduced new dynamic-behavior use cases and deployment scenarios into network operator labs around the world to test multi-vendor interoperability of the T-API 2.0 NBI. The 2018 SDN T-API interoperability demonstration builds on the OIF's 2016 interoperability test and demonstration which addressed multi-layer and multi-domain environments as well as on the 2014 demo which prototyped the use of Northbound APIs and helped advance transport SDN standardization.

The 2018 multi-vendor interoperability demo was led by global network operators in Asia, Europe and North America on lab- and cloud-deployed systems testing T-API service enablement functionality for connectivity, topology and notification services in combinations that can provide the necessary functionality to automate service fulfilment and service assurance. The APIs were tested against four dynamic use cases including: 1) multi-level connection establishment; 2) multi-domain restoration; 3) multi-layer reroute; and 4) multi-NFVI interconnection.

The demo also incorporated service provisioning scenarios at the MEF LSO (Lifecycle Service Orchestration) Presto reference point in the MEF LSO architecture, using the MEF NRP Interface Profile Specification (MEF 60), which defines T-API extensions in support of MEF Carrier Ethernet services.

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. The work on T-API 2.0 has addressed the main issues that were identified in OIF's 2016 Interoperability Demonstration and communicated by OIF to the industry. Some additional functional as well protocol related issues and gaps were identified in the latest round of testing using additional use cases and deployment scenarios. These are described in greater detail in the report. The experiences of the testing will be shared across the industry to help develop critical implementation agreements and specifications. The benefits of the demo experience and on-going collaboration with ONF and MEF include: 1) advancing industry convergence on SDN control approaches; 2) validation of T-API operational improvements and service enhancements; and 3) laying the groundwork for autonomous networking.

The OIF, in collaboration with MEF, has helped establish a foundation for open, programmable networks that allow operators to efficiently deliver dynamic multi-domain connectivity services to the market.

2 Introduction

Network operators are deploying 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:

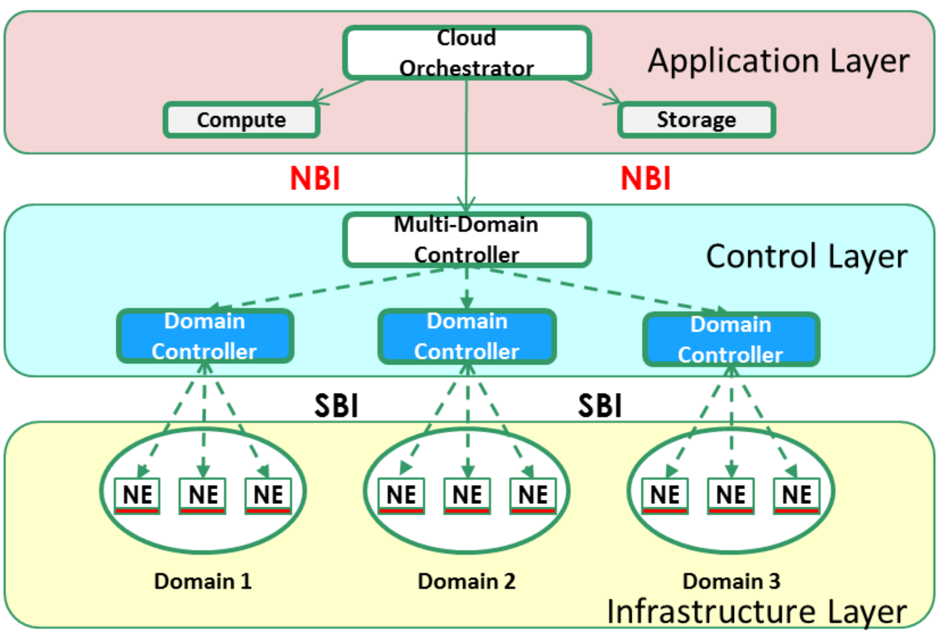

Figure 1: SDN framework for multi-domain operator networks (Source: ONF)

3 Demo Setup

3.1 Worldwide Test Topology

Testing was carried out in 4 network operator labs in Asia, Europe and North America as shown below in Figure 2:

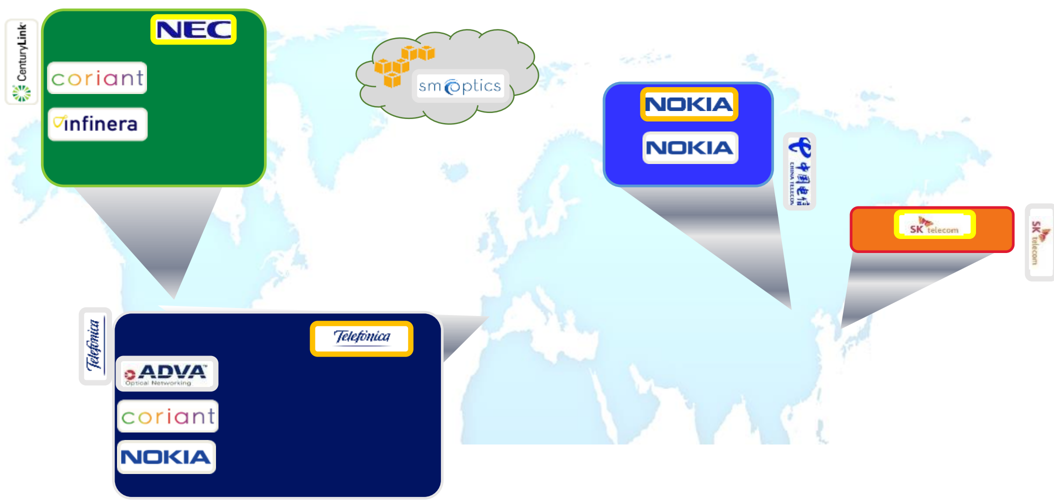

Figure 2: Worldwide Test Topology

Participating Network Operators:

- Asia: China Telecom, SK Telecom
- Europe: Telefonica
- North America: CenturyLink

Consulting Network Operator:

- TELUS

Six vendors and one research institute participated with software or hardware, providing a variety of types of equipment and software functions. The testing involved lab- and cloud-deployed systems. One consulting operator also provided support and monitored results. The participating vendors included:

- ADVA Optical Networking
- Coriant
- Infinera
- NEC Corporation/Netcracker
- Nokia
- SM Optics

Research Institutions:

- Centre Tecnològic Telecomunicacions Catalunya

3.2 Testing Methodology

The 2018 Demonstration largely followed the testing methodology used in previous OIF SDN testing. The use cases tested focused more on dynamic network conditions and the testing procedures were modified to incorporate testing of MEF specifications and allow participation of cloud-based implementations as well as implementations physically located in the carrier lab.

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. Testing was also carried out in parallel between some specific carrier labs and cloud implementations of T-API using the common signaling control network.

Finally, in the inter-lab phase, testing was conducted between Multi-Domain Controllers and Domain Controllers in different carrier labs and in the cloud 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 Technical Specifications

4.1 Transport API (T-API)

T-API (Transport API) is a standard API defined by the Open Networking Foundation (ONF) that allows a T-API client, such as a carrier's orchestration platform or a customer's application, to retrieve information from and control a domain of transport network equipment controlled by a T-API server such as a Transport SDN Controller.

The initial T-API 1.0 specification and SDK was published by ONF in 2016 and included the following:

<u>Topology Service -</u> supports retrieval of Topology information from the Controller in the form of Node, Link & Edge-Point details.

<u>Connectivity Service-</u> allows the client to retrieve information about and request new point-to-point, point-to-multipoint and multipoint-to-multipoint connectivity service across the transport network.

Notification - allows the client to subscribe to and filter autonomous notifications from the server for events such as resource or service state changes, failure or degradation.

Path Computation - allows the client to make a request for Computation & Optimization of paths.

Virtual Network Service - allows the client to create, update, delete Virtual Network topologies.

4.1.1 T-API 2.0 Enhancements

T-API 2.0 was initially published by ONF in late 2017 and updated as version 2.0.1 in 2Q2018 [T-API].

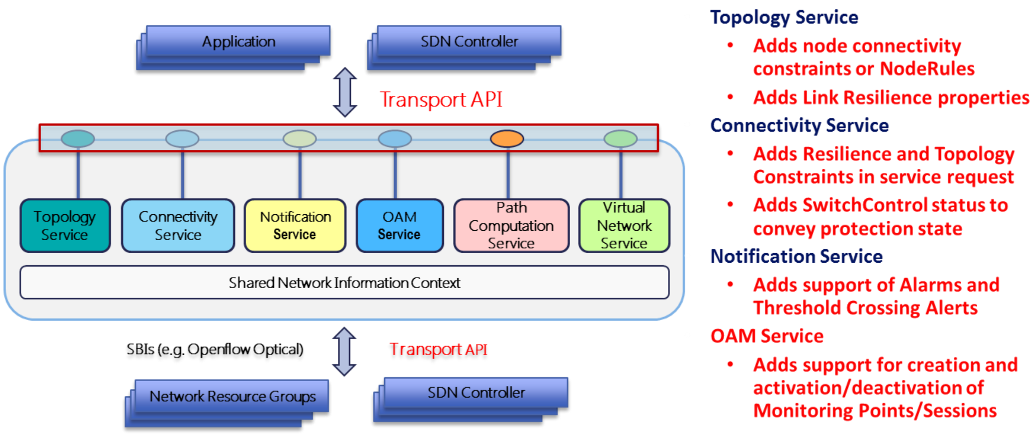

Figure 3: T-API 2.0 Enhancements

T-API 2.0 enhancements include the following:

Corrections and Alignments

During 2016 interop testing in the joint OIF/ONF T-API Demonstration, several corrections and alignments to the current usage of YANG were noted, such as the format for attribute lists, the use of lisp-case as opposed to camel-case for YANG objects, support for extensible enumeration and other minor corrections. Based on the corrections the T-API 2.0 YANG model has now been tested and validated by multiple YANG compilers for correctness.

Additionally, the Spec model has been improved with both simplifications and enhancements, and there have been some naming and refactoring updates to improve the overall T-API information model.

Finally, several naming changes and extensions to the model have been made based on joint discussion with MEF to avoid creating issues with MEF terminology and services. One main example is that the T-API 1.0 ServiceEndPoint (SEP) has been renamed ServiceInterfacePoint (SIP) in T-API 2.0.

Node Connectivity Constraints and metrics

One finding of the 2016 testing was that the initial model did not provide detailed enough information for computing connectivity service path across nodes, especially any port-to-port connectivity constraints or metrics associated with a node. There has always been an option to expose a detailed sub-topology within a node using the same node and link constructs, however this may in some cases add unwanted overhead and complexity, and in other cases may not be sufficient to express certain types of constraints.

Instead T-API 2.0 adds the ability to define Rules relating ports or NodeEdgePoints within the Node, as expressed in a set of NodeRuleGroups. These NRGs may identify Rules affecting Forwarding, Capacity, Cost, Timing or Risk, and express constraints between the use of NEPs associated with a NodeRuleGroup. In addition, it is possible to express constraints on forwarding between NRGs using an InterRuleGroup.

In the most direct form, it is possible to specify an NRG between each pair of ports in the Node, expressing a simple MAY or CANNOT forwarding rule or additionally Cost or Capacity constraints on forwarding between the ports.

Autonomous notification for updates and telemetry

T-API 1.0 provided basic support for autonomous notifications from the controller to client to indicate changes of state. T-API 2.0 enhances these capabilities by defining alarms and threshold crossing alerts. Alarms can be qualified by their perceived severity, cause and whether they are service-affecting, while threshold crossing alerts can be qualified by the threshold parameter and associated value.

Resilience and Protected Connections

The resilience model focus is the Switch (CIM FcSwitch) construct which represents the forwarding selector[1] and which enables changes of forwarding to achieve resilience. The model also represents the control element of the resilience control loop that monitors behavior, assesses that behavior identifying necessary configuration changes and applies those configuration changes to make the required adjustments to forwarding to achieve the intended resilience.

OAM

Finally, T-API 2.0 introduces OAM services as a feature. The T-API 2.0 model is extended to incorporate Maintenance Entities, Maintenance Entity Groups, Maintenance End Points and Maintenance Domain Intermediate Points according to standard ITU-T and MEF definitions that allow the client to determine where monitoring points may be present as well as to start, terminate, enable and disable measurement services between specified points in a connection. OAM services are a critical component of providing service which meets service level agreements, as well as supporting fault localization and isolation when a fault is discovered.

4.2 MEF NRP

The T-API model and SDK is being leveraged by MEF in the Network Resource Modeling (NRM) and Network Resource Provisioning (NRP) projects where they are extending T-API with MEF-specific extensions [NRM/NRP]. MEF plans to then test, demonstrate and certify these APIs as part of the MEF OpenCS reference implementation projects (e.g. Optical Transport and OpenCS Packet WAN).

MEF Presto NRP focuses on network activation across the MEF LSO (Lifecycle Service Orchestration) Presto interface with SOF (Service Orchestration Functionality) client to ICM (Infrastructure Control and Management) server bi-directional communication. In addition, topology-based retrieval of components, including those that apply to a given Connectivity Service are defined.

MEF NRP is aligned with the ONF T-API model. ONF T-API model is a simplification of the ONF Core IM to make this more appropriate for transfer over an interface and make the terminology more familiar to users with experience in transport network modeling.

It standardizes a single core technology-agnostic specification that abstracts common transport network functions. ONF T-API capabilities can be extended through the *specification* pattern. The essential approach is to associate an instance of a ONF T-API *entity class* with a set of constraints and/or extensions that account for the specific case (*specification class*).

These *specification classes* are definitions of specific cases of usage of an *entity class*, enabling machine interpretation where traditional interface designs would only allow human interpretation. MEF NRM is designed as a set of *specification classes* which extends, or augments, ONF T-API defined *entity classes*.

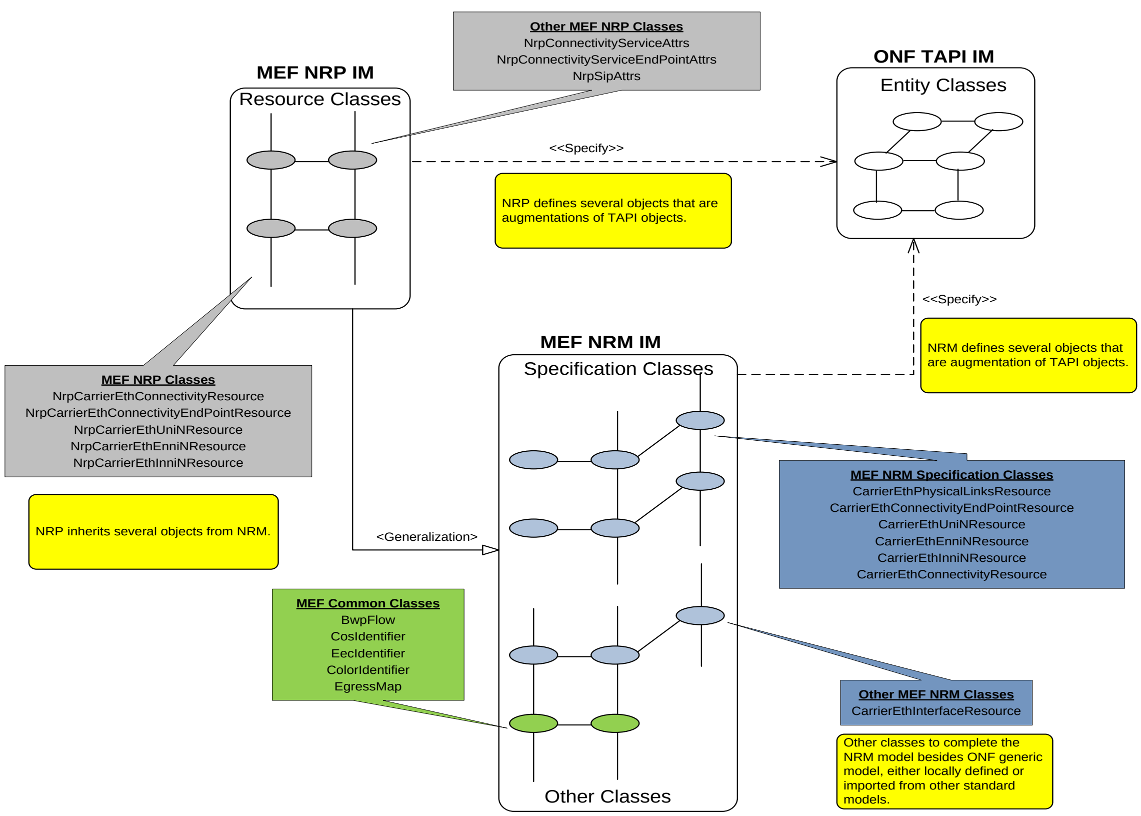

Figure 4: ONF T-API/NRM/Presto NRP Model Relationships

Topology functionality is a key component of the Presto NRP solution. Presto NRP clients can leverage topology retrieval functionality for multiple Presto NRP scenarios. For example, during an augmentation of an existing Presto NRP supporting service topology retrieval should be used prior to adding a resource (i.e., UNI for MEF service) to verify the resource will not break the existing service. Topology can also be leveraged for temporal check pointing of the network resources and path used to support an existing MEF services.

The collaboration between ONF T-API, NRM and Presto NRP begin with leveraging the Connectivity and Topology Services from ONF T-API to NRM/Presto NRP. Presto NRP leverages NRM resource activation model and provides necessary UML artifacts that are intended to be leveraged for YANG data model mappings.

The future implementation of other ONF T-API constructs such as Path Computation and Notification Service will provide augmentations to Presto NRP. In addition, a similar network activation and topology specification interface profile needs to be provided for OTN (Optical Transport Network). A major benefit of this functionality and API will be for overlay/underlay control and management. Specific to overlay/underlay are activation and topology support of a device or set of devices that support both Ethernet and OTN interfaces.

5 TESTING T-API SERVICE ENABLEMENT FUNCTIONALITY

The international joint-network operator, multi-vendor optical networking vendor testing will test, verify and examine the T-API 2.0 functionality. Specifically, the Connectivity, Topology and Notification Services defined with T-API are examined. These software patterns are essential in the management and control of optical and hybrids of optical and other technologies.

The T-API common model and corresponding northbound API between SDN Controllers and Orchestrator are beneficial to carriers and vendors. Without a common model and set of APIs, each carrier would have to work with their respective vendors to implement proprietary model and APIs. Vendors would end up having to work with multiple carriers to implement carrier specific models and APIs. This situation is expensive from both a monetary and time perspective.

Three specific APIs and patterns are examined with the testing – connectivity, topology and notification. Each of these patterns is valuable by itself. However, using these patterns in combinations can provide the necessary functionality to automate service fulfilment and service assurance. In a multi-domain implementation, the Orchestrator will also be responsible for Path Computation which is also a T-API service.

Service fulfillment is the combination of Connectivity and Topology Services. The Orchestrator will initiate the process with the service decomposition. The result of service decomposition is a graph or topology of available paths to meet the service request. The Orchestrator will use perform a path computation across the set of domains and send a Connectivity Service request to the two or more SDN Controllers. Each SDN Controller and network will make a respective decomposition and corresponding path computation and perform a corresponding Connectivity Service activation of network resources. Test cases specific to Connectivity, Topology and Path Computation are examined.

Service assurance is the combination of Notification and Topology Services. During the lifecycle of a Connectivity Service faults will potential occurs. These faults will result in topology changes within the one or more domains. The Notification Service is essential for set of SDN Controllers and Orchestrator to remain in synchronization with the underlying network. Test cases specific to topology changes within the network and corresponding T-API defined Notifications are tested and examined.

5.1 Topology and Connectivity Services

The MEF LSO (Lifecycle Service Orchestration) architecture is used as a layered and abstract architecture for the OIF testing. Topology exists at every layer within a defined architecture and framework. Topology is represented using graph concepts (i.e., vertices and edges). A customer service is graph and is represented as a topology with node edge points and links. The service topology is known at the Business Application layer.

The customer service can be decomposed into the underlying network resources that are used to provide the connectivity for the service. The network resource connectivity is constructed of one or more connected topologies. The decomposed service topology is known at the Service Orchestration Functionality layer.

The one or more topologies at the SOF layer can be further decomposed at the ICM layer with finer granularity of topologies and down to nodal level. The nodal level can be physical and/or virtual. The OIF SDN Interoperability testing will test and verify the implementation of the T-API Topology Service. The Topology Service API will provide a network synchronized topological representation of the individual domains as well as cross domain topologies.

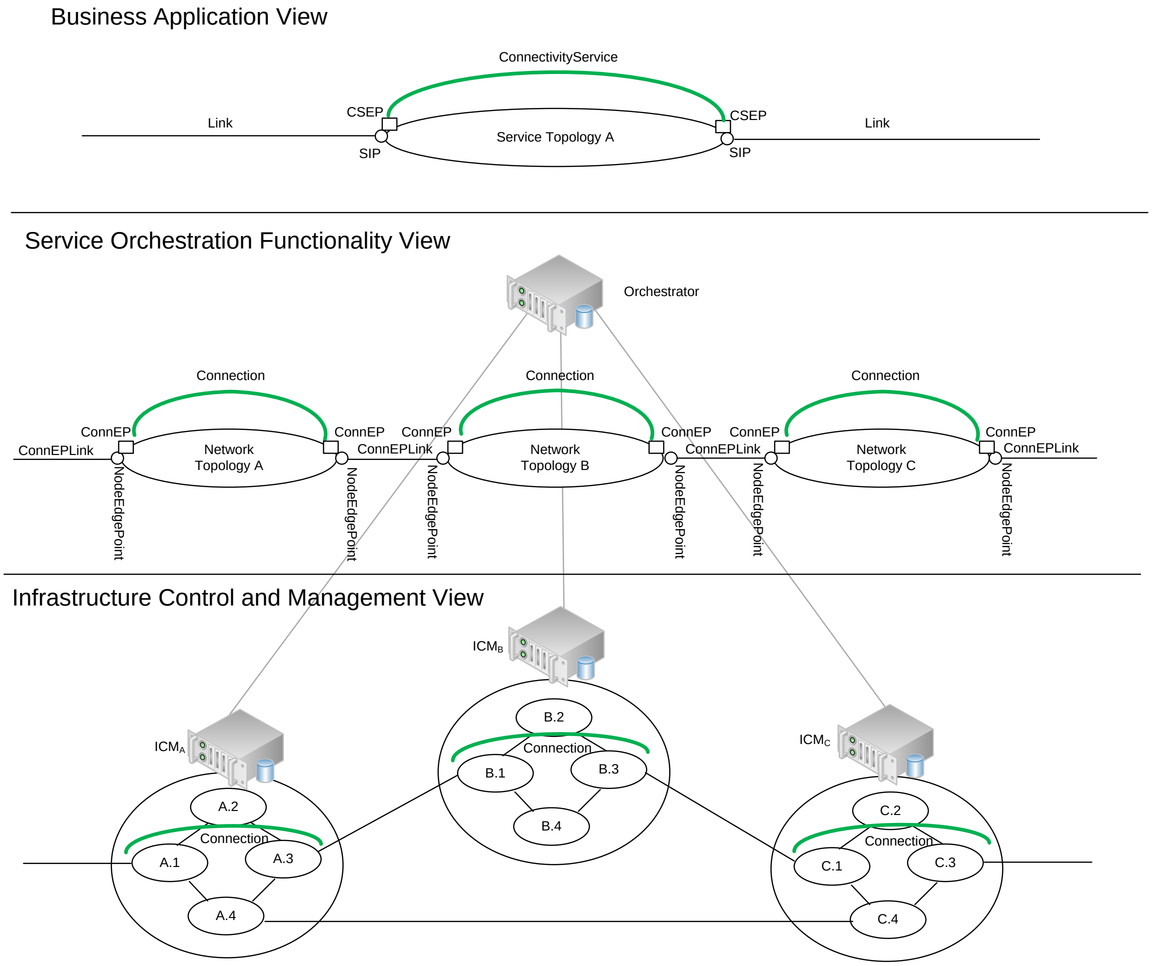

Figure 5: Topology and Connectivity Services

5.2 Topology and Notification Services

T-API northbound APIs enable the Orchestrator and corresponding SDN Controllers to stay synchronized with the state of each domain and specific set of devices within each domain. The Orchestrator can stay topologically synchronized with the under domains by subscribing to Topology Notifications. In the event of a Topology change the SDN Controller will publish a Notification to subscribers (i.e., Orchestrator). In the case of a multi-domain network this gives the Orchestrator the ability to reroute Connectivity Service that may be impacted across a set of two or more domains.

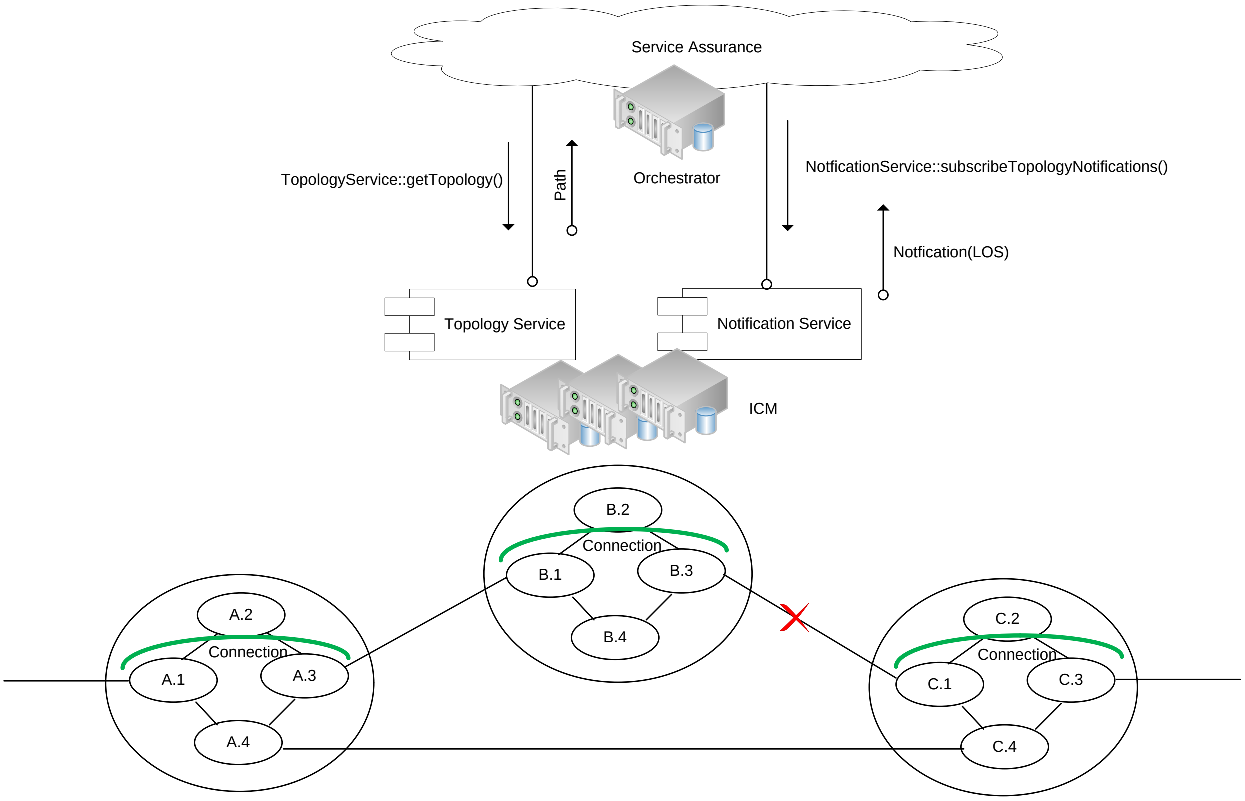

Figure 6: Topology and Notification Services

5.3 Other Epics of T-API Patterns

The following section details the set of Epics that leverage the various T-API patterns and corresponding APIs.

5.3.1 Business Application Service Visualization

T-API patterns are applicable at more than just the Orchestration to SDN Controller interface. Topology, Connectivity, Path Computation and Notification Services are equally applicable up the stack such as at the Business Application Layer. A Service Fulfillment application can leverage T-API to provide BA layer views of Connectivity Services to internal and external customers.

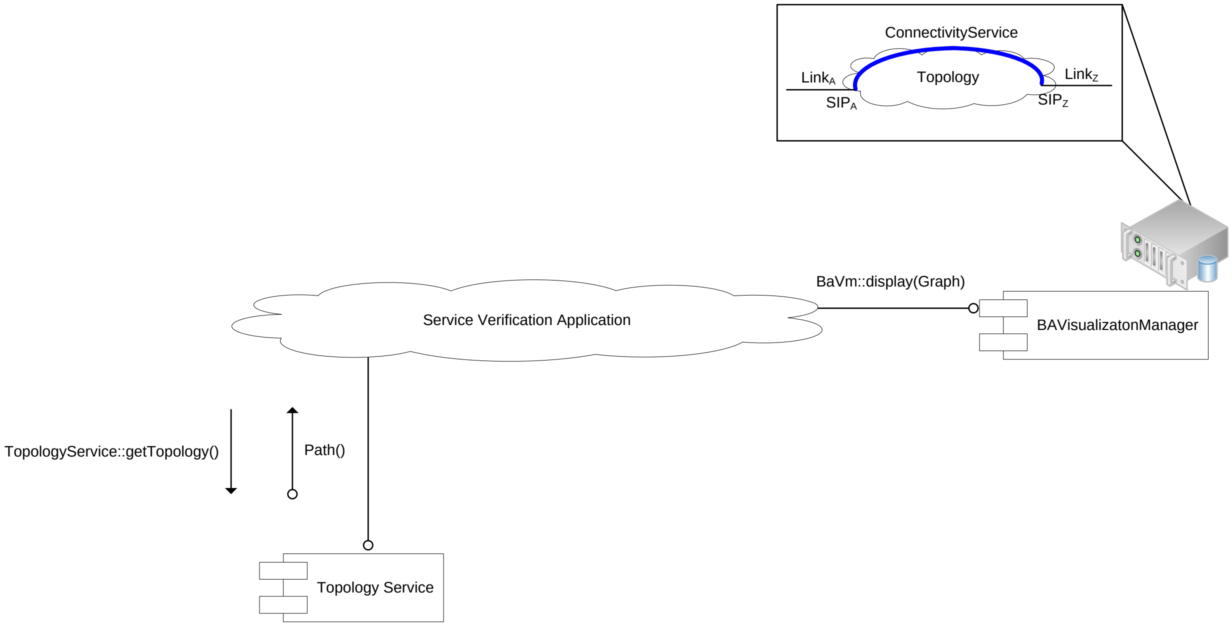

Figure 7: Business Application Connectivity Service Epic

5.3.2 Network Resource Visualization

The ability to scope and filter the topology provides a scalable approach to finding the specific layer of topology for service fulfillment and service assurance functions.

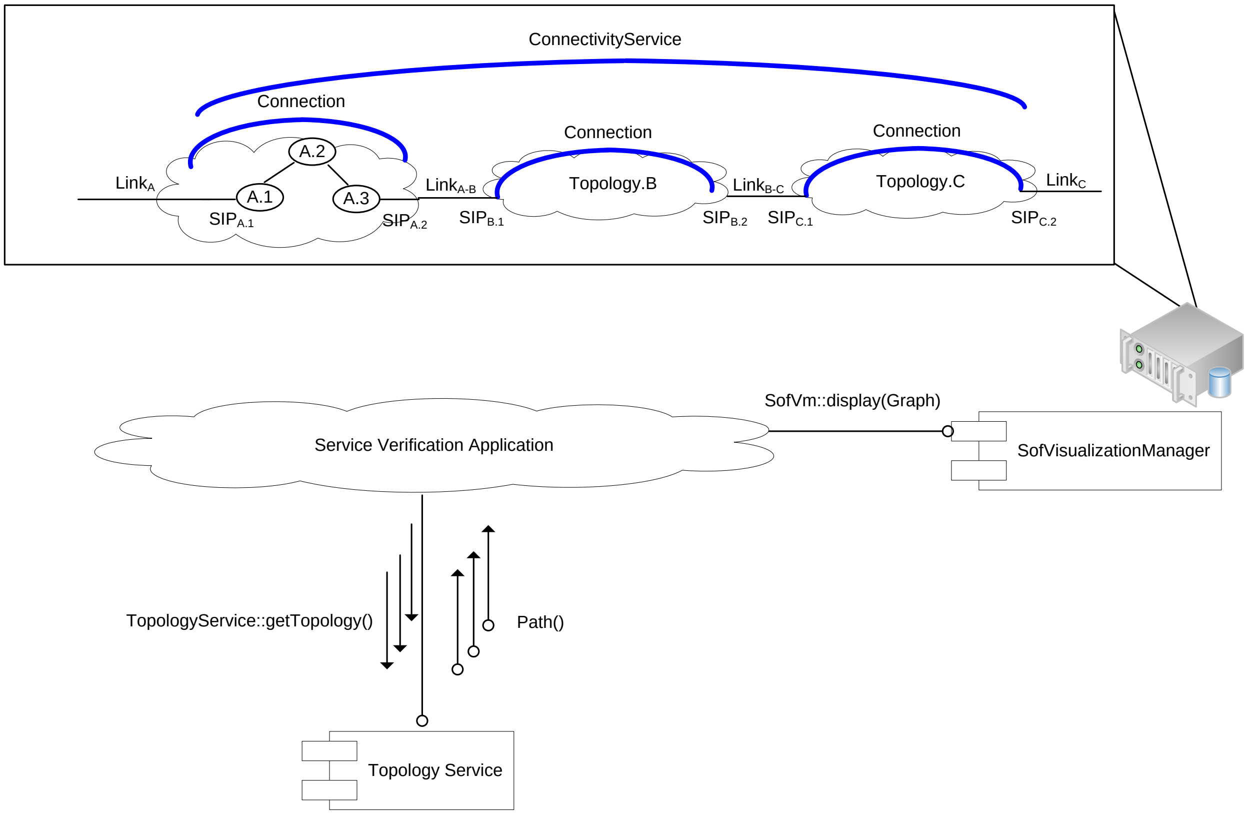

Figure 8: Network Resource Epic

6 USE CASES

6.1 MULTI-LAYER CONNECTION ESTABLISHMENT

In the basic multi-layer connection establishment use case, T-API is used as in the figure below to retrieve topology from domain controllers in a multi-domain, multi-layer network and provision connectivity services end-to-end across multiple domains.

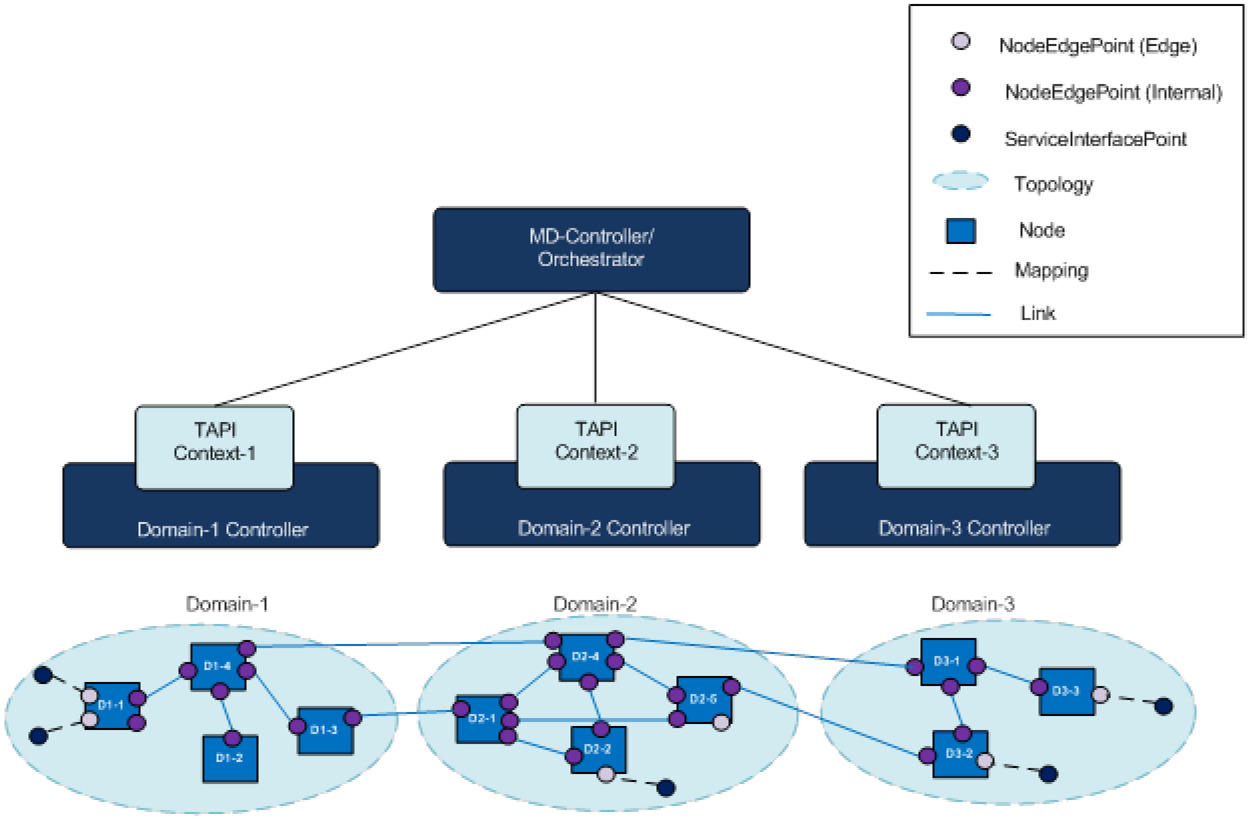

Figure 9: Connection Establishment

The MD-Controller/Orchestrator uses the T-API Topology service first to retrieve link and node details for the domains (GET Topology) and correlates these with its knowledge of inter-domain connectivity. Depending on the desired level of control the MD-Controller/Orchestrator can compute just the inter-domain path and leave the intra-domain details to each Domain Controller or it can compute the detailed path to be used within each domain and specify this in the command to provision the service (POST Connectivity Service). Path constraints can also be specified such as diversity requirements using Shared Resource Link Group (SRLG) information from the topology.

Since the full OIF testing topology incorporates multiple laboratories, as well as cloud-based Domain Controllers, there are multiple stages of testing, initially between MD-Controller/Orchestrators and individual Domain Controllers within a lab and then across multiple labs using the inter-lab SCN network.

6.2 Multi-Layer Reroute

Rerouting traffic is a network operations task that needs to be done in cases where traffic patterns or network situations are changing. Examples are specific traffic flows at sports events, network congestions in some network areas or on some links, or scheduled maintenance. Network restoration is another key reason for traffic rerouting.

Rerouting can be performed in a single network layer such as IP, Ethernet or the photonic layer. This, however, can create a risk for suboptimal resource allocation. Networks can be operated much more efficiently with multi-layer rerouting, taking a holistic view of all network layers and optimizing resource utilization across multiple network layers.

The risk of suboptimal rerouting decisions is highlighted by a simple, common example: traffic needs to be rerouted from an ingress node to a new/different egress node. It might happen that both routers are in the same network region with a short physical distance (maybe located in two neighboring cities). Let's further assume that both routers are not directly connected, but via core routers. In this case, it might happen that the IP layer calculates a relatively "long" path for connecting both routers as it is not aware of physical topology information. It could make sense from an IP layer perspective given the current link metric situation. It would, however, result in many transit routers along the path between ingress and egress node. All these routers need to be equipped with physical ports providing IP capacity for forwarding the transit traffic.

Studies have shown that significant router capacity and ports could be saved in networks if IP routing is aware of transport topology (multi-layer rerouting). There are two key contributions to this saving. On the one hand, the number of transit routers could be decreased if physical topology is known and shorter physical paths would be selected. On the other hand, optical bypass could be introduced where all or some of the transit routers could be avoided.

Telefonica conducted a related study a couple of years back. They analyzed the entire network in Spain. One of the key results showed a saving in the number of IP ports of up to 37% through multi-layer network optimization. This result was calculated for a multi-layer restoration scenario, which is one form of a rerouting use case.

Today multi-layer rerouting is a manual process. The IP team of a network operator must request support from the operational team. This team needs to analyze the capability of the optical network (e.g., available optical links, bandwidth/wavelength utilization on these links, …) and send a response back to the IP team, including a time schedule when requested optical connectivity could be provided. Such a manual process can take several days (or even weeks).

In a software-defined networking scenario, this process would be automated (future mode of operation). A network orchestrator receives topology information from all network layers and domains. With the optical layer, it would get information from the optical SDN controller via T-API (including metrics like SLRG and link latency). Typical network orchestrators support multi-layer network optimization functions. Such a function would calculate various path options in all layers and recommend optimal IP and optical path combinations in case of a reroute request. Some orchestrators even conduct such multi-layer network optimization in an off-line mode, enabling optimal path combinations to be recommended at short notice, avoiding long processing times.

As mentioned, scheduled network maintenance is another use case where a multi-layer reroute capability is required. Examples are node or link updates, repairs or replacements. Here network operators need to identify all services affected during such a maintenance procedure. Although maintenance work is typically related to a single network domain or layer (e.g., optical layer), it does also impact other layers (e.g., the IP layer). Therefore, multi-layer rerouting should be conducted by SDN orchestrators. In the case of maintenance work in the optical layer, the related process could be triggered via T-API and domain-specific SDN controllers.

Scheduled maintenance can be divided into node and link maintenance. In the first case a service needs to be rerouted to a different path, avoiding the affected node. There are different options for this process:

a) Remove existing services -> remove network element -> establish new services when topology is refreshed in domain SDN controller
b) Remove existing services -> establish new services using exclusion path (old node is then used as a service constraint)
c) Modify all services using exclusion path before removing old node (old node is used as a service constraint)

When maintenance work is finished, the network operator may want to reroute services back to the original route. This can be accomplished through the following additional steps:

d) Remove existing services -> establish new services using inclusion path when new node is available (node is then used as a service constraint),
e) Modify all services using inclusion path when new node is available (node is used as a service constraint)

The second case, link maintenance, when for instance a fiber cable that needs to be repaired. Following options could be applied:

a) Remove existing services -> remove fiber -> establish new services when topology is refreshed in domain SDN controller
b) Remove existing services -> establish new services using exclusion path (original link is then used as a service constraint)
c) Modify all services using exclusion path (original link is used as a service constraint)

The very last option, including modification, is the most save one from a service operations perspective as it covers removal and creation in a single revertible transaction. What becomes very important for multi-layer networks are overlay umbrella orchestrators. Such platforms have got the full knowledge which services would be affected by an outage of a selected link and can coordinate multi-layer rerouting. The following procedures could be applied as soon as the repaired fiber is available again:

d) Remove existing services -> establish new services using inclusion path (original link is then used as a service constraint)
e) Modify all services using inclusion path (original link is used as a service constraint)

6.3 Multi-domain restoration

The ONF SDN Architecture enables a network to be divided into multiple domains. This could be to help with network scaling, having domains setup to reduce the overall number of NEs being controlled by a federation of SDN controllers; to separate equipment supporting different technologies, where one domain could be Optical, one TDM electrical and one Packet; to align with administrative boundaries, where individual domains would align with planning/operations group responsibilities; or to align with broad geographic boundaries, where domains align with different continents. Whatever the reason for dividing the network into domains, restoring services should still be possible if allowed by the Service's SLA for the network resources available. But with the network divided into domains, there are scenarios where domain-oriented restoration will not be able to recover faults. The two primary scenarios are: 1) a domain may not have adequate resources to recover a failed service, and 2) a link interconnecting two domains may be the location of a fault.

This is where Multi-Domain restoration is required.

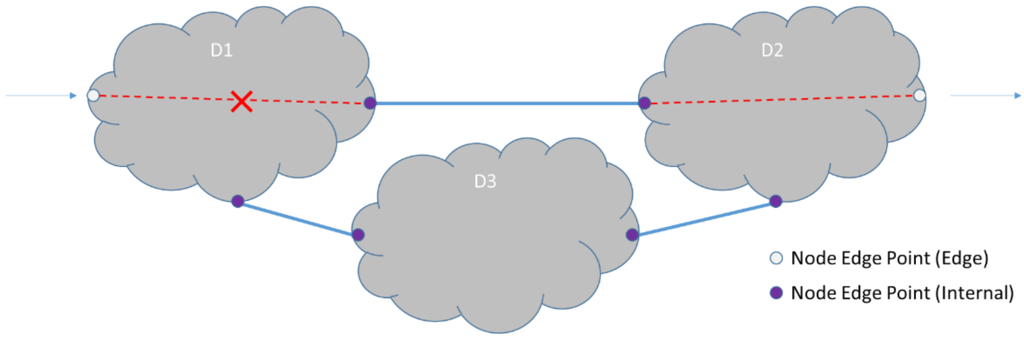

Figure 10: Domain internal fault

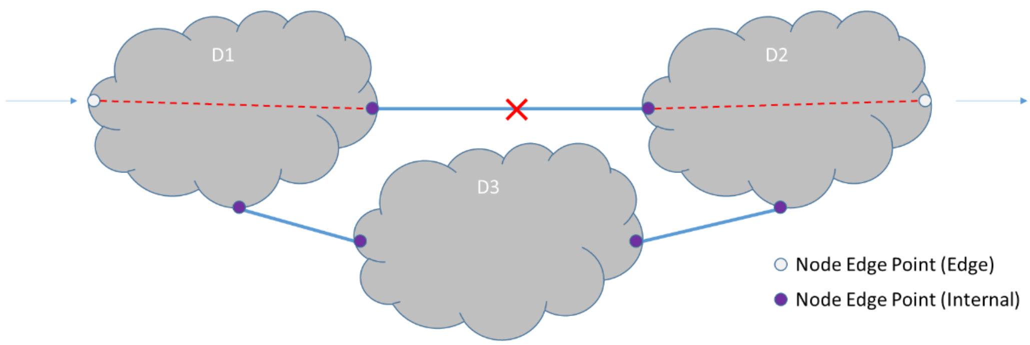

Figure 11: Inter-domain link fault

Multi-Domain restoration requires the network orchestrator to coordinate the actions of two or more domains to recover a service. An example restoration process consists of the following steps:

a) Orchestrator receives a trigger indication to start the restoration action. This would be either a service faulted-nonrestorable indication from a domain or an indication that a link between domains has faulted.
b) Orchestrator determines the path across the orchestrator's domain of operation to restore the service. This is similar to the determination of a path for new service establishment. The result is a list of links between domains and domain connections to be used to provide the service. To reduce the amount of resources required by the restoration process, some of the links and domain connections may already be in existence and used by the current, failed service.
c) Orchestrator initiates new domain connections and modifies existing connections where necessary. Modification would usually be changing the Node Edge Point used at the domain boundary for a connection already existing in the domain. Note: The Orchestrator may choose to retain the resources used by the faulted connection, allowing the service to return to its original path once the failure has been resolved. This means modification does not necessarily

remove the original Node Edge Point or release the resources in use as a part of the modification request.

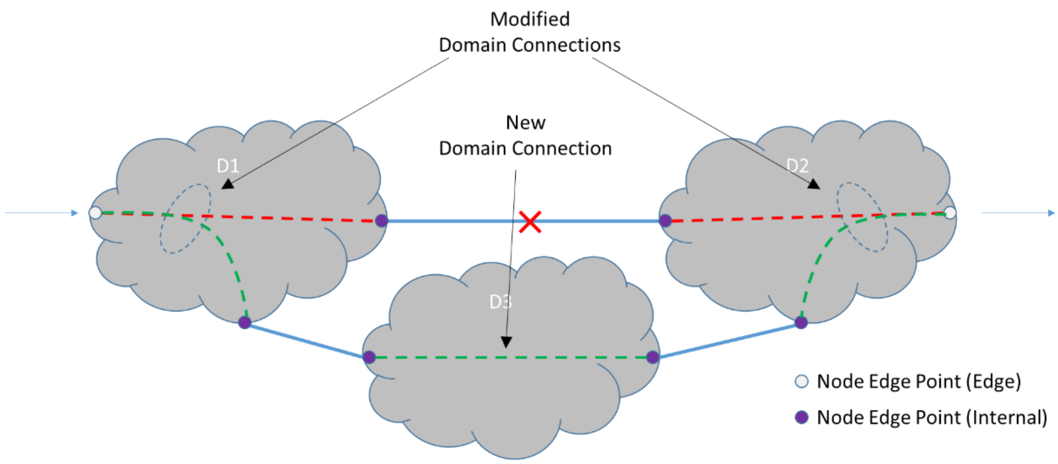

Figure 12: Use of Modified Domain Connections

d) Orchestrator receives an indication the new domain connections and modified connections are complete and resumes monitoring Service OAM.
e) Orchestrator receives an indication the service fault or link fault have been cleared by original reporting domain(s), allowing for the service to revert to its original path.
f) Orchestrator waits for reversion criteria (e.g. wait-to-restore, restoration schedule) to be met and then modifies domain connections to return traffic to original path. New domain connections that were established specifically to restore the service may now be released.

As shown, the support multi-domain restoration is dependent on notifications from domain controllers and the ability to modify existing connections. Both capabilities are new with TAPI 2.0. As a result, TAPI 2.0 enables high availability delivered by restoration across multiple domains.

6.4 Multi-NFVI Interconnection

NFV Infrastructure Points of Presence (NFVI-PoP) are typically distributed among several data centers that are interconnected through different transport network using different technologies. The 2016 OIF-ONF T-API interoperability event demonstrated the need for connectivity service instantiations between different NFVI-PoPs for the Network Service Life Cycle Management [OIF16]. [IFA022] presents several use cases to support network services deployed on top of infrastructure which is interconnected over a Wide Area Network (WAN) infrastructure. In this context, a Network Service, which is a composition of Virtual Network Functions, is instantiated by the interactions among OSS/BSS, NFVO, WIM/VIM, and Network Controllers. Figure 13 shows the proposed architecture in [MAN001].

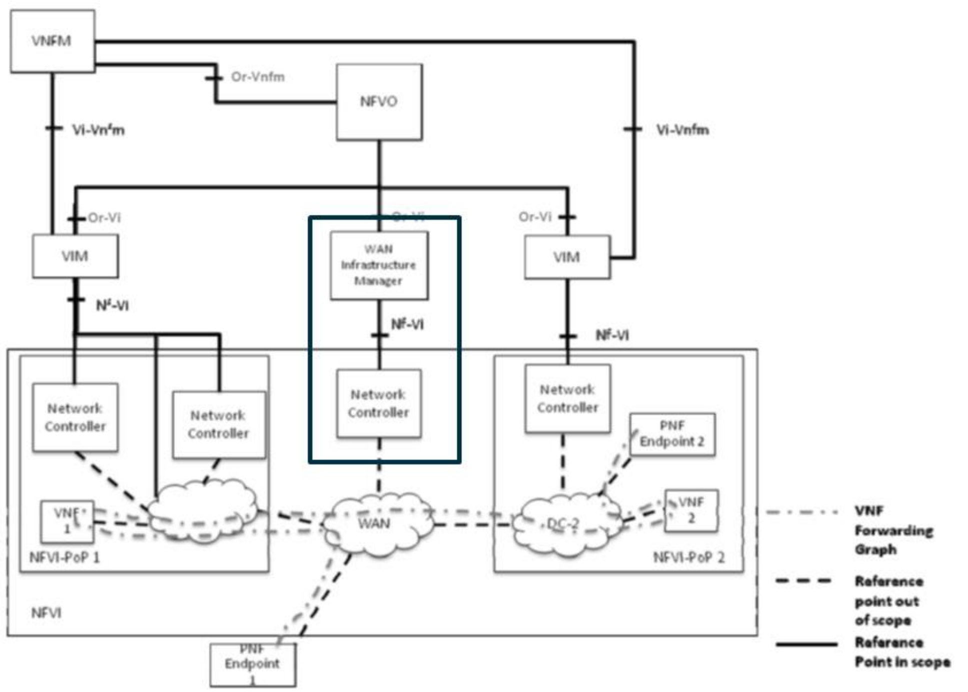

Figure 13: Network service deployment through multiple NFVI-PoPs

In view of the above considerations, the objective of this use case is to demonstrate a connectivity life cycle management with multiple SDN-based Network Controllers over WAN interconnections that act as a WIM. As shown in Figure 14, the target scope of this use case is architecturally configured by a set of network controllers, acting as a WIM, and in a multi-site environment are several Wide Area Networks (WAN) infrastructure. The WAN architecturally interconnects multiple NFV sites and supports connectivity services. This use case will provide valuable experiences that will enable to better provide the connectivity service towards the NFVO.

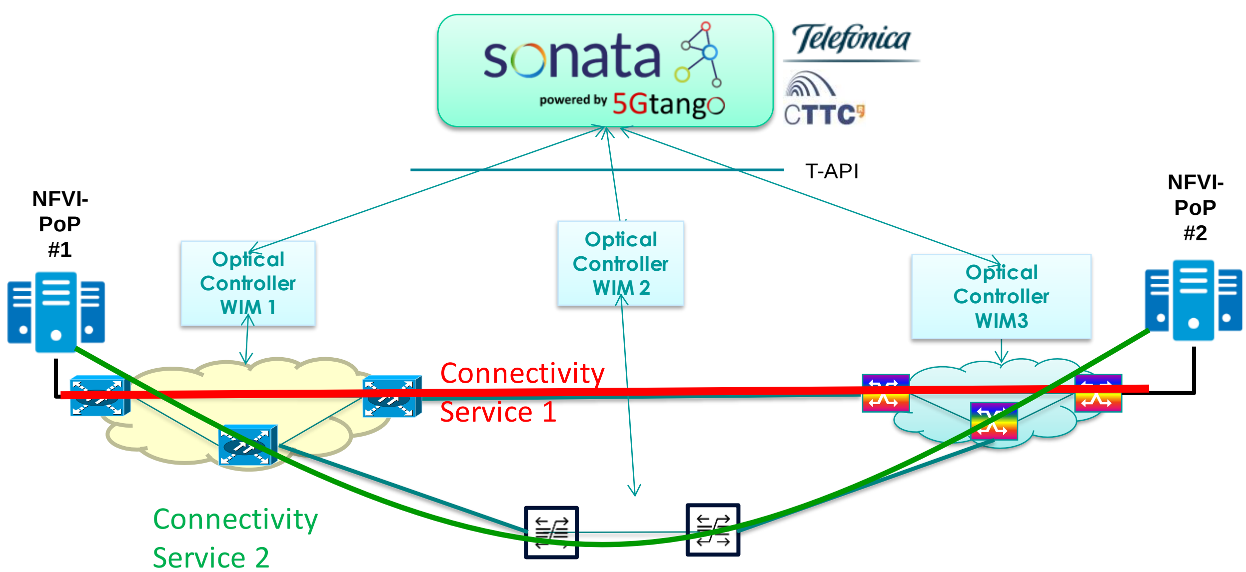

Figure 14: Multi-NFVI interconnection use case

To support this use case, we have used SONATA NFV Platform, which is an Open Source NFV Management and Orchestration Framework [SON18]. SONATA NFV Platform is a service programming, orchestration, and management framework. It provides a development toolchain for virtualized network services, fully integrated with a service platform and orchestration system. It introduces a modular and flexible architecture for the Service Platform (NFVO) and discuss its main components and features, such as function- and service-specific managers that allow fine-grained service management, slicing support to facilitate multi-tenancy, recursiveness for improved scalability, and full-featured DevOps support.

For the implementation of the use case, Transport API (T-API) implements a set of northbound application interfaces of the network controller. Based on select use case from [IFA022], topology and connectivity services shall be implemented and tested. The topology service can be used when WIM explores a set of connectivity end points whose requirements may be specified from OSS\BSS and\or NFVO (e.g. affinity group, location constraint). The topology service can also be used when WIM may collect WAN QoS parameters (e.g. capacity, latency, cost, etc.) that shall satisfy the Network service QoS requirements as imposed by the OSS/BSS and\or NFVO.

The connectivity service can be initialized when WIM executes path instantiation. Two or more end points given by the WIM are then interconnected with the connectivity service. The NFV-O might include several constraints to the requested connectivity service. In this use case, T-API 1+1 redundant Connectivity service provisioning is explored. Another explored topic is the necessity of requesting specific transport labels from the NFVO perspective. Connectivity service extensions to support these novel features might be needed to be considered in upcoming T-API releases.

7 FINDINGS

Testing in the 2018 Demonstration 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. The work on T-API 2.0 has addressed the main issues that were identified in OIF's 2016 Interoperability Demonstration and communicated by OIF to the industry, especially:

- issues with YANG consistency;
- need for additional node connectivity attributes;
- inclusion of notification for synchronization of setup between MD and Domain Controllers; and
- use of consistent RESTCONF style across participants.

Several new issues were identified through testing of additional use cases and focus on dynamic behavior in the 2018 testing:

- Consistency of LEAFREFs with RESTCONF specifications
    - this has been addressed already by ONF in T-API 2.0.2.
- Multiple approaches to domain abstraction used in different Domain Controllers
    - The ONF’s SDN architecture expects controllers will necessarily utilize different abstractions when representing the real topology being controlled, based on differences

in business requirements as well as technology. Inclusion of more examples of abstraction may be helpful for implementors of MD Controllers.

- Need for additional restoration operational controls
    - While basic restoration is now supported in T-API 2.0, additional controls as OIF has previously defined in [OIF-ENNI-REC-AM-01.0] would be helpful additions.
- Bidirectional relationships
    - There was some disagreement among participants on this issue: some felt that making more relationships bidirectional would improve scalability and performance while others felt that this would add significant duplication of information and processing overhead to maintain the reverse relationships.
- URL usage
    - The use of LEAFREF and UUID in T-API requires the application to know the object hierarchy in the model; it was suggested that JSON-LD style references could be an improvement – however these are not consistent with RESTCONF. This issue was not resolved.
- Need for better error description through HTTP result codes
    - Result codes currently identified in the specification do not provide detailed information that could be used for debugging purposes, and more specific codes should be added as a future work item.
- Methods for scaling of notification
    - It was noted that the notification method results in significant load on the Domain Controller, and that scaling methods such as notification hubs should be considered in future implementation.
- Containment models for equipment
    - The relationship between Endpoint Identifiers in the API and actual ports in the equipment may differ between vendors, and a translation function would be useful – this is being investigated as a work item in ONF T-API 2.1.
- Automation of inter-domain link discovery
    - Currently inter-domain links are assumed to be pre-configured in the MD Controller as an individual Domain Controller only sees one end of the link. Using an automated discovery mechanism such as are identified in [OIF-ND-IA-01.0] would reduce the chance of human error and simplify configuration.

8 BENEFITS

The interest of major service providers and technology suppliers on open and standardized transport control interfaces builds on the common understanding that manual processes neither meet operational cost nor agility requirements imposed by real-time provisioned cloud resources. What`s more, the applicability of SDN goes far beyond addressing specific needs of present networks and adding some new use cases, as it is an essential enabling technology for transforming networks from manual to automated and finally fully closed-loop, autonomous operation.

The joint Transport API interoperability demo is advancing transport networking in various dimensions: (1) maturing and hardening the SDN technology for deployment in production networks (2) creating a basis for immediate operational cost reductions as well as extending the service portfolio and (3) paving the way towards autonomous networking.

8.1 ADVANCING SDN CONTROL

The need for open, standardized control interfaces has triggered action with various standard bodies and fora. There was some concern that approaches were uncoordinated and fragmented. With MEF joining the collaboration between OIF and ONF, backed up by a strong community of major service providers and well reputed suppliers, there is now a strong foundation for a single common approach to SDN. Aligning the work of standardization bodies and industrial alliances reduces the diversity of interfaces. This helps service providers to efficiently integrate their networks. The testing across different networks verifies the enhanced T-API and MEF Presto specifications and has identified some further improvement. Combining the skills and experience of service providers and vendors assures a close alignment of technical feasibility with operational requirements.

8.2 OPERATIONAL IMPROVEMENTS AND SERVICE INNOVATION

Any additional capability provided by a control interface enables a communication service provider to save cost by replacing a manual task with an automated process. The functionally extended T-API 2.0 interface improves visibility of performance information and resource availability as well as resource restrictions to a northbound controller, enabling a network operator to move manual tasks into an orchestration layer. Adding MEF expertise improves service end-point modelling. This is an important step towards practical implementation of automated lifecycle management, including service activation and assurance.

8.3 FOUNDATION FOR AUTONOMOUS NETWORKING

Deep learning (DL) and artificial intelligence (AI) are technologies to augment and replace human reasoning with software-initiated decision-making by analyzing and processing a comprehensive set of network data. DL/AL can be used to predict failing devices, recommend optimization strategies, identify anomalies, and forecast user demand. This will help service providers to run the network more efficiently and fully autonomously with closed-loop control.

AI/ML is key to making this happen, but open network control for automated activation and notifications for efficient capturing of network information are also key enabling technologies. Recent additions to T-API and Presto have significantly enhanced the scope of information provided through these interfaces and have extended the telemetry streaming / notification capabilities.

Rolling out AI/ML technologies as software on central servers might take a matter of days, while the implementation of network interfaces could easily become a multi-year network evolution as network elements might need to be upgraded and new capabilities may need to be added. A stable, well tested and feature-rich transport network control and notification interface is essential for a seamless transition from manual to automated and finally autonomous operations. The joint work with the transport API interop demo is an invaluable asset, providing a secure foundation for this operational transformation.

9 CONCLUSION

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. The work on T-API 2.0 has addressed the main issues that were identified in OIF's 2016 Interoperability Demonstration and communicated by OIF to the industry. Some additional functional as well protocol related issues and gaps were identified in the latest round of testing using additional use cases and deployment scenarios. The experiences of the testing will be shared across the industry to help develop critical implementation agreements and specifications. The benefits of the demo experience and on-going collaboration with ONF and MEF include: 1) advancing industry convergence on SDN control approaches; 2) validation of T-API operational improvements and service enhancements; and 3) laying the groundwork for autonomous networking.

The OIF, in collaboration with MEF, has helped establish a foundation for open, programmable networks that allow operators to efficiently deliver dynamic multi-domain connectivity services to the market.

10 APPENDICES

10.1 LIST OF CONTRIBUTORS

- Dave Brown (Editor), Nokia
- Pawel Kaczmarek, ADVA Optical
- Ulrich Kohn, ADVA Optical
- Victor López, Telefónica
- Arturo Mayoral, Telefónica
- Konrad Mrowka, ADVA Optical
- Stephan Neidlinger, ADVA Optical
- Lyndon Ong, Ciena
- Jack Pugaczewski, CenturyLink
- Jonathan Sadler, Coriant
- Ricard Vilalta, CTTC

10.2 ABOUT THE OIF

The OIF facilitates the development and deployment of interoperable networking solutions and services. Members collaborate to drive Implementation Agreements (IAs) and interoperability demonstrations to accelerate and maximize market adoption of advanced internetworking technologies. OIF work applies to optical and electrical interconnects, optical component and network processing technologies, and to network control and operations including software defined networks and network function virtualization. The OIF actively supports and extends the work of national and international standards bodies. Launched in 1998, the OIF is the only industry group uniting representatives from across the spectrum of networking, including many of the world's leading service providers, system vendors, component manufacturers, software and testing vendors. Information on the OIF can be found at http://www.oiforum.com.

10.3 T-API Supplement

T-API is a standard NorthBound Interface for a Transport SDN Controller. It supports both high level technology- independent service (i.e., intent-like) and detailed technology-specific service, depending on policy. As discussed below, T-API has also been adopted by other industry SDOs and forums for their specific needs.

As a component of Transport SDN, T-API enables programmatic control of the carrier's transport network to support faster and more flexible allocation of network resources to support application demands (e.g., bandwidth or latency). The benefits include reduction of cost due to operational simplification and reduced delay for the introduction of new equipment and services, as well as the ability to develop and offer new revenue-producing services such as network slicing and virtualization for 5G and IoT applications.

Some of the unique benefits of T-API include:

- T-API supports unified control of domains with different technologies using a common technology-agnostic framework based on abstracted information models. This allows the carrier to deploy SDN broadly across equipment from different vendors and with different vintages, integrating both greenfield and brownfield environments as opposed to requiring major turnover and investment in new equipment.
- T-API is based on telecom management models that are familiar to telecom equipment vendors and network operations staff, making its adoption easier and reducing disruption of network operations.
- T-API combines both standards specification development and open source software development, providing code for implementation and testing that allows faster feature validation and incorporation into vendor and carrier software and equipment.

10.3.1 History

The T-API project was initiated at ONF in 2014, following a successful multi-vendor multi- carrier prototype demonstration of Transport SDN architecture jointly held with OIF. A determination from the testing was that the NBI presented a critical interface for SDN deployment across domains with different equipment and capabilities.

At the same time a technology-independent Common Information Model (CIM) project in ONF provided a framework for the T-API information model built on many years of experience in development of management interfaces at TMF, ITU-T and other SDOs. T-API was agreed to be a purpose-specific, use-case-driven Interface Profile Specification of the ONF Core Information Model that would reflect the concepts, patterns and evolution of the Core IM.

10.3.2 Design/Framework

The work on CIM and T-API is part of an overall vision to develop interfaces and APIs for SDN that are based on an "invariant" model, i.e., one that is independent of the protocols and technologies in current fashion and can be flexibly mapped to protocol using model-based automated code generation rather than costly, time-consuming handcrafting of interface software. This approach aims to avoid creation of "software silos" that could be restrictive to operators just as vendor-based silos are today.

The T-API model is accordingly defined in the same manner as the CIM, using the Unified Modeling Language (UML). UML is a well-known, stable standard (ISO) and is familiar to people with experience in transport network management standards, as well as many other industries and sectors close to telecom. Use of UML allows an easy bridge to ITU-T, TMF, MEF and NFV management modeling work – it should be noted that UML is also in use in some IETF groups.

UML is supported by open source tooling which allows graphical presentation of the models. The two-dimensional representation in UML makes objects and especially the relationships between objects easier to visualize than a linear text specification such as a YANG model. Tooling also supports constriction of consistent views that ease explanation and understanding

UML is protocol-independent and can be mapped to different data schema languages and associated wireline protocols. UML provides similar descriptive properties to data modeling languages such as YANG, including specification of object classes, superclasses and stereotypes, attribute types, rw vs. ro characteristics, ability to transfer attributes by reference, interfaces, operations and notifications. A multi-organizational effort has been underway to specify the mapping from UML to YANG to make the mapping process capable of being done via software rather than being a manually intensive process [ISOMII].

10.4 REFERENCES

[OIF16] OIF SDN Transport API Interoperability Demonstration (February 2017), http://www.oiforum.com/documents/sdn-transport-api-interoperability-demonstration/

[T-API] ONF Transport API Software Development Kit (SDK) , https://github.com/OpenNetworkingFoundation/TAPI

[NRM/NRP] https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_59.pdf and https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_60.pdf

[IFA022] Network Functions Virtualization (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services ETSI GR NFV-IFA 022 V3.1.1 (2018-04).

[MAN001] ETSI GS NFV-MAN 001: "Network Functions Virtualization (NFV); Management and Orchestration".

[SON18] Sonata NFV Platform, http://5gtango.eu/

O. Gerstel, C. Filsfils, T. Telkamp, M. Gunkel, M. Horneffer, V. López and A. Mayoral: Multi-Layer Capacity Planning for IP-Optical Networks, in IEEE Communications Magazine Feature Topic "Advances in Network Planning". January 2014, Vol. 52, Issue. 1, pp. 44-51. DOI

[OIF-ENNI-REC-AM-01.0] Recovery Amendment to E-NNI 2.0 Common Part (February 2014), http://www.oiforum.com/wp-content/uploads/OIF-ENNI-REC-AM-01.0.pdf

[OIF-ND-IA-01.0] Neighbor Discovery Implementation Agreement (August 2015), http://www.oiforum.com/wp-content/uploads/OIF-ND-IA-01.0.pdf

[ISOMII] https://wiki.opennetworking.org/display/OIMT/IISOMI

[CIM] ONF Open Information Modeling & Tooling Project,
https://wiki.opennetworking.org/display/OIMT