OPEN NETWORKING
FOUNDATION

Transport API (TAPI) 2.0 Overview

Version 0.0
August 13, 2017

DRAFT

ONF Document Type: White Paper
ONF Document Name: TAPI 2.0 Overview

Disclaimer

1 TAPI Overview

1.1 introduction

TAPI (Transport API) is a standard API defined by the Open Networking Foundation (ONF) that allows a TAPI 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 TAPI server such as a Transport SDN Controller. In other words, TAPI 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, TAPI has also been adopted by other industry SDOs and forums for their specific needs.

As a component of Transport SDN, TAPI 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 TAPI include:

- TAPI 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.
- TAPI 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.
- TAPI 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.

1.2 Potential TAPI Use Cases

Potential applications of TAPI include many opportunities to integrate control and monitoring of the optical transport network with higher level applications, such as:

- End-to-end dynamic bandwidth services across a multi-domain carrier network with support of resiliency and reoptimization
- Interconnection of multiple CORD (Central Office Rearchitected as a Datacenter) sites across a carrier's multi-domain network
- Support of Virtual Transport Network services offering dynamically controllable and monitorable virtual resources connecting remote sites of large customers
- Support of network slicing enabling connectivity for high bandwidth or ultra-low latency 5G services by isolated and secure virtual subsets of the network

1.3 History

The TAPI 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 TAPI information model built on many years of experience in development of management interfaces at TMF, ITU-T and other SDOs. TAPI 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.

1.4 Design/Framework

The work on CIM and TAPI 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 TAPI 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].

2 TAPI Software

2.1 TAPI SDK

The TAPI Software Development Kit includes the base UML model as well as corresponding models, specifications and software, all made publicly available through the ONF Open Source SDN github site for the SNOWMASS software project[1]. The SDK consists of:

- The TAPI UML Information Model, derived from the ONF Core Information Model using a pruning and refactoring process, to conform to the ONF Transport API Functional Requirements Specifications (TR-527)
- The YANG Data Schema, generated from the UML model using OSSDN EAGLE project guidelines & generation tools
- The Open API Specification (OAS/Swagger), generated from YANG schema using OSSDN EAGLE project generation tools & RESTConf specification
- The Python code stubs for the Reference Implementation, generated from the Open API specification (OAS/Swagger) using OSSDN EAGLE project generation tools, and then populated with a thin mapping layer to ONOS API and an associated example network created using the Mininet network emulator.

The automation tools developed by the ONF OpenSource SDN EAGLE project [EAGLE] map UML specifications into YANG data schema as an automated process, and similarly map YANG into JSON Swagger specifications and Swagger into python or java code stubs. The result is no longer dependent on hand-coded development of YANG and thus less likely to result on errors in the YANG specification. Moreover, since the base specification is in UML, changes to the core features of YANG (which is still an evolving standard) can be managed through changes to the automation tools rather than having to manually rewrite all of the YANG documentation.

Together with the CIM, the set of automation tools in Eagle was developed in ONF to automate the process of mapping UML specs to YANG models and from YANG models to JSON Schema. The underlying UML models provide a simpler and more easily understood model structure, while tools provide a consistent and reliable method of generating YANG. The SDK development process will ultimately make it possible to generate prototype code rapidly and easily and make the process of revision and extension faster and simpler.

2.2 TAPI Development Process

To realize a software-centric approach to standardization, TAPI follows an agile work methodology which in turn means shorter release cycles, with smaller feature sets in each release, increased focus on model and code artifacts and interleaving of the releases with targeted implementation efforts to validate and identify shortcomings.

For each TAPI release, work items are prioritized based on purpose specific use cases to satisfy the needs of targeted interops, PoCs and other partner SDOs TAPI-based initiatives. This

[1] https://github.com/OpenNetworkingFoundation/Snowmass-ONFOpenTransport

impacts the set of features supported in a particular release, potentially resulting in fewer features being added initially as the features are gated by the need for implementations rather than a pure paper spec. Note that while specific TAPI releases are scoped to satisfy a set of chosen use cases, the purpose of TAPI always has been to support new use cases and technology advancements, rather than limit itself to current network technologies.

2.3 Extensibility

The ability to extend the model (both within ONF and by external groups) without conflict and with clear definition and identification is an important feature for future evolution and usability. In TAPI as in the CIM, extensions/augmentations can be created at any level of the model, using the specification (Spec) model. The Spec model essentially uses a composition approach to extend/augment the model (as opposed to inheritance). The composed parts are controlled by the specification definition[2] where the attributes of the specifications will be augmented with stereotypes that direct their usage.

Extensions to the API can be defined by other SDOs and industry bodies or by individual vendors according to their particular needs or special features. The MEF, for example, is in the process of defining extensions to the API based on its service definitions for Ethernet Services. These extensions will form an MEF technology-specific Spec model for MEF Ethernet services.

3 TAPI 2.0 Features

3.1 Basic Features from TAPI 1.0

The TAPI 1.0 specifications and SDK supported the following features:

3.1.1 Topology Service

The Topology Service supports retrieval of Topology information from the Controller in the form of Node, Link & Edge-Point details. This information can be used for path computation, planning and analysis purposes and supports virtualization of network resources for particular client applications

3.1.2 Connectivity Service

The Connectivity Service 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. Support for both single layer and multi-layer connectivity services is included. A connection is the provisioned potential for forwarding (circuit/packet) between two or more Edge Points of a Node.

3.1.3 Notification

The Notification Service 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. Autonomous notifications are carried via a transport mechanism such as websockets.

[2] Some use is still made of conditional composition but it is expected that the specification approach will be used to drive all conditional content.

3.1.4 Path Computation

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

3.1.5 Virtual Network Service

The Virtual Network Service allows the client to create, update, delete Virtual Network topologies. The client initially requests support of a traffic matrix describing traffic requirements between client Service Interface Points, and the controller responds by allocating the necessary resources within the customer's virtual network.

3.2 TAPI 2.0 Enhancements

TAPI 2.0 enhances TAPI 1.0 with corrections and alignments based on 2016 interop testing as well as extensions of the model to support more complex connection path computation, additional notification capabilities, connectivity service resilience and OAM services.

3.2.1 Corrections and Alignments

During 2016 interop testing in the joint OIF/ONF TAPI 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 TAPI 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 TAPI information model.

Finally, a number of 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 TAPI 1.0 ServiceEndPoint (SEP) has been renamed ServiceInterfacePoint (SIP) in TAPI 2.0.

3.2.2 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 TAPI 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 particular 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.

Using NodeRuleGroups for groups of ports and supplementing this by InterRuleGroups it may be possible to express constraints in a more compact way or alternatively to express more complex or overlapping constraints.

3.2.3 Autonomous notification for updates and telemetry

TAPI 1.0 provided basic support for autonomous notifications from the controller to client to indicate changes of state. TAPI 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.

3.2.4 Resilience and Protected Connections

TAPI 2.0 also introduces support of connection resilience using protection and restoration. The resilience model utilizes the Switch construct (mapping to the CIM FcSwitch construct) which represents the forwarding selector and enables changes of forwarding between alternative paths 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 recovery behavior.

3.2.5 OAM Services

Finally, TAPI 2.0 introduces OAM services as a feature. The TAPI 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 TAPI Status and Testing/Implementation History

4.1 TAPI Status

TAPI 1.0 was released in August 2016, its release being timed to align with a 2016 joint OIF/ONF interop demonstration. As the demonstration progressed, feedback from implementation and testing was incorporated into updates to the models and prototype code as interim versions TAPI 1.0.x.

TAPI 2.0, the next major revision of TAPI, has been tagged as Release Candidate in August 2017 with target for finalization in September, and incorporates both updates from interop testing as well as additional features that were either identified as needs during testing or recognized previously as missing but not ready for incorporation for the 2016 testing, e.g., OAM and protection.

TAPI will continue to follow a plan of multiple releases per year as new features are incorporated and updates made based on testing results.

OIF 2014 demo

As mentioned above, the Transport SDN model and NBI concepts were first implemented and tested in a prototype demonstration held in 2014 jointly by OIF and ONF. This demonstration included 5 supporting carrier labs and 9 vendors. A white paper with details of the testing and results is available at the OIF website (http://www.oiforum.com)

OIF 2016 demo

The 2016 demo incorporated the formalized standards for TAPI and the TAPI Software Development Kit and involved 5 supporting carrier labs + 2 consulting carriers and a total of 11 participating vendors, including orchestration/multi-domain controller systems provided by both vendors and participating carriers. Results have also been documented in a joint OIF/ONF white paper that is available at the OIF website (http://www.oiforum.com)

Individual PoCs and field trials

Telefonica/CTTC are continuing to participate in follow-up testing with their vendors, and supporting associated research projects that have been featured at OFC and other venues

In 2016, China Telecom carried out an SDN controlled OTN+Ethernet network field trial based on TAPI in Quanzhou, Fujian, China. In 2017, CT expects to continue this field trial with more nodes and functions added.

4.2 Adoption by other industry bodies

MEF

The TAPI model and SDK is being leveraged by MEF in the Network Resource Modeling (NRM) and Network Resource Provisioning (NRP) projects where they are extending TAPI with MEF-specific extensions. 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).

OIF

As discussed above, OIF has adopted the TAPI specifications as the basis for interoperability testing of Transport SDN solutions in 2016 and is pursuing further interop testing in 2018.

ITU-T

ITU-T has aligned its protocol neutral information model for the control plane view (ITU-T Recommendation G.7718) with the CIM work in ONF as the basis for modeling of transport networks for control purposes.

5 Further Information

Additional information can be found in the following sites:

TAPI (General)

TAPI Architecture

SNOWMASS Open Source Software Project

ONF Common Information Model (CIM)

IISOMI

TAPI 1.0

TAPI 1.0 Functional Requirements

TAPI 1.0 Software Development Kit (SDK version 1.0.3)

TAPI 2016 Testing Report

TAPI 2.0

TAPI 2.0 overview presentation

TAPI 2.0 Software Development Kit (RC1)