SDN Framework and APIs

Lyndon Ong
OIF Marketing Committee Co-Chair
Ciena

OFC 2016
March 22, 2016
Multi-Domain Transport SDN Model

Multi-Domain Integration

Transport SDN framework for carrier networks

- Used to unify diverse carrier domains
  - multiple technology layers
  - multiple domains with differing control planes
  - greenfield and brownfield

- Need for standards on application layer interface to control layer (SDN NBI)
Framework for SDN APIs

- Opening up access to control components
  - Call/Connection Control, Topology, Path Query, Virtualization
  - Replace internal, proprietary interfaces, decouple functions/SW
Transport SDN Framework and APIs

- **Focus on work in OIF Transport SDN Framework and joint work between OIF and ONF on Transport API**

  - **OIF Network & Operations Working Group**
    - **Objective**: facilitate the development of interoperable networking and operations solutions for multi-technology networks
    - **Leadership**: Peter Landon, BTI, Chair

  - **OIF Interoperability Working Group (Network)**
    - **Objective**: define and carry out proofs of concept multi-vendor interoperability trials of OIF Implementation Agreements
    - **Leadership**: Jonathan Sadler, Coriant, Chair

  - **OIF Carrier Working Group**
    - **Objective**: develop requirements and guidelines for the services and functions to be supported by future optical networks
    - **Leadership**: Vishnu Shukla, Verizon, Chair

  - **ONF Open Transport Working Group**
    - **Objectives**
      - Develop SDN and OpenFlow® standard-based control capabilities for carrier transport networks.
        - Recent change: addition of Wireless Transport project
    - **Leadership**: Lyndon Ong, Ciena, Chair
    - **Work to date**
      - Transport SDN Use Cases & Functional Requirements
      - OpenFlow Extensions for Optical Transport
      - 2014 Joint Demo with OIF
    - **In Progress**: T-API, Information Model & OpenFlow v1.1
Achieving Common APIs
The Tools and Remaining Challenges

Existing Work
- Current API work is being done in fragmented silos
- Some linkage of APIs to existing protocol environments

Keys to achieving interoperable common APIs
- Base work on common Information model and API specification
  - Take advantage of ONF Common Information Model project - aligns ONF, ITU, TMF, MEF, OIF
- Verify APIs provide the necessary functionality
  - Use case review and convergent SDO work
  - Refinement for transport network applications
  - Prototype, demonstrate, implement!
Common Information Model

- Defines a common object model for all types of Software Defined Networks
  - Basic components like network resources, service constructs

- Common agreements on modeling across SDOs
  - ONF, ITU-T, TMF, MEF...

- Apply Transport requirements to Common Info Model to create Transport API (TAPI)
Transport API Model

- Can be hierarchically applied – Parent controller to Child controller
OIF Transport API Project Overview

Collaborative Effort with ONF
- Develop Use Cases and Functional Requirements
  - Basis of work
- Information Model
  - Based on and extends ONF Core IM
- Data Models/Schema
  - YANG model and JSON schema
  - https://github.com/OpenNetworkingFoundation/ONFOpenTransport
- Implement, test, refine – “agile” process

Software and Automation Tools
- Englewood Open Source SW project
  - https://github.com/OpenNetworkingFoundation/ENGLEWOOD
- Eagle ONF Open Source Tools project

OIF Interop testing and IAs to follow
## Connectivity Service Functional Requirements (draft)

<table>
<thead>
<tr>
<th>TAPI_FR_0001</th>
<th>Create Connectivity Service</th>
</tr>
</thead>
</table>
| **Description** | • Causes creation of a *Forwarding-Construct* representing the *Service* request to connect the *Service-End-Points* within the shared *Context* between API Client and Provider  
• Returns Service ID to be used as reference for future actions  
• Initial definition will be for a basic point-to-point bidirectional service  |
| **Pre-conditions** | • Requestor/Client has visibility of the set of *Service-End-Points* between which connectivity is desired within the *Context*  
• Requestor/Client has information about the types of connectivity available and constraints it can specify such as Service Level  
• Requestor/Client may be aware of other existing Connectivity *Services* and their IDs  |
| **Inputs** | • List of *ServiceEnds* and details of each including  
  – Role of the terminating *ServiceEndPoint* in the context of the *Service*  
  – Directionality of the terminating *ServiceEndPoint* in the context of the *Service*  
  – Reference (Name/ID) to terminating *ServiceEndPoint*  
• Connectivity Requirements such as Layer and Capacity  
• Connectivity Constraints such as Latency, Cost, etc  
• Start Time & End Time  |
| **Outputs** | • Service ID  
• Operational State  
• Lifecycle State  
• Confirmation of Service Characteristics : See above inputs  |
| **Notifications** | Success/Failure  
Change of Operational State  |
| **Error-conditions** | Service not supported  
Service input not supported  
Endpoint not recognized  |
| **Post-conditions** | Oif – cite specific documents  
Onf  
IETF  |
| **Sources** | [https://github.com/OpenNetworkingFoundation/ONFOpenTransport](https://github.com/OpenNetworkingFoundation/ONFOpenTransport)  |

---

---
Model of data plane resources in an SDN-enabled network

- Technology agnostic
- Recursive (Forwarding Domain may contain FDs)
- Models static and dynamic elements
- Extensible to different technologies and environments
Future Work

- **Implement and Demonstrate**
  - OIF/ONF Demonstration

- **Develop OIF Implementation Agreements**
  - Select options from base TAPI spec in ONF
  - Specify formats and encoding agreements

- Iterate with more experience and use