Open Transport Switch:
Supporting SDN in Transport Networks

Ping Pan ppan@infinera.com
Wes Doonan WDoonan@advaoptical.com
Lyndon Ong Lyong@Ciena.com
Geoffrey Mattson gMattson@juniper.net
Meral Shirazipour meral.shirazipour@ericsson.com
Young Lee leeyoung@huawei.com
Thomas Nadeau tNadeau@juniper.net
Saurav Das saurav.das@bigswitch.com
Piotr Myslinski pmyslinski@vellosystems.com

October 2012
Acknowledgments and Other Contributors

• Inder Monga (Esnet)
• Martin Swany (Indiana University)
• Matthew Davy (Internet2 / Indiana University)
• Guy Roberts (DANTE)
• Richard Hughes-Jones (DANTE)
• Lars Fischer (NORDUnet)
• Jerry Sobieski (NORDUnet)
• David McDysan (Verizon)
Problems in Transport Networks

1. Typically, transport equipment cannot talk to each other outside one network domain.

2. Network operation over multiple layers (including ROADM, OTN, Ethernet and MPLS) is very expensive.

3. Service deployment is very slow (due to technology, operational practice and politics).
SDN Means Different Things to Different People

• Application Providers: business incentives
  – Faster service deployment
  – Simpler network connection provisioning
  – Better network resource utilization

• Network Providers: reality check
  – Control the distribution of network interior information
  – Need to integrate multi-layer/multi-regional networks
  – Continue the use of dynamic protocols for provisioning and protection in some parts of the network

We need to take the above into consideration to enable fast adaption, development and deployment in transport networks
Why SDN?

• Overlay
  • Deploy services on top of network infrastructure
  • Operate over multiple networks, over multiple layers
  • Technology-agnostic

• Virtualization
  • Enable the control of relevant nodes and links
  • Enable the optimization of network resources
  • Provide needed security and isolation
SDN for Transport Networks is not hard

This is where SDN services are defined and offered

This is where SDN API’s need to be defined to support configuration, monitoring and provisioning.
How to Build an Overlay?

- SDN Controller
- Transport Infrastructure
- Open Transport Switch
- Access Devices
- Overlay Network

A virtual switch built on top of transport equipment for overlay interfacing.
Open Transport Switch:
A light-weight virtual switch in transport equipment

SDN Controller

XMPP/others
JSON /others
OpenFlow+

OTS-Management Agent
OTS-Control Agent
OTS-Data Agent

Control Plane

Internal Communication

Data Forwarding Plane

Transport Switch Platform

Physical Interfaces

For Open Discussion
Key Functions

• OTS Data Agent
  – Responsible for setting up data-plane connections, such as cross-connects
  – Use OpenFlow and additional extensions (see later)

• OTS Control Agent
  – Responsible for switch configuration (Controller → OTS)
  – Responsible for exporting network topology (OTS → Controller)
  – Protocol Choice
    • JSON is one solution, based on our experience in ALTO and OVS
    • Other protocols (e.g. XMPP) would also work
Key Functions (cont.)

• OTS Management Agent
  – Responsible for real-time event data (e.g. alarm) exchange (OTS → Controller)
  – Responsible for service negotiation and discovery at boot-up time (OTS ↔ Controller)
  – Protocol Choice
    • XMPP is a simple and reliable way for event messaging
    • DNS may be used for service discovery and negotiation

• OTS representation
  – A single node: such as an optical switch
  – A couple of nodes: such as an optical switch + a router
Operation Example #1: Explicit Provisioning

Topology report:
- L1: 10GE
- L2: to B, 100GE

Provisioning (e.g., OpenFlow):
L2-L3 :: map VLAN-200 packets to VIF X

For Open Discussion
Operation Example #2: Implicit Provisioning

Provisioning:
• Setup A-B with BW X

Provisioning:
• Setup C-D with BW X
• Map data to C-D

For Open Discussion
The existing OpenFlow needs work...

<table>
<thead>
<tr>
<th>Functions</th>
<th>Example</th>
<th>Problem</th>
<th>Possible Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection setup</td>
<td>F1 (ODU1) - Interface 1 ⊗ Interface 2 - F2 (ODU1)</td>
<td>What if the flows are non-packet (e.g. waves); it’s not defined in OpenFlow</td>
<td>Introduce the concept of flow-id into OpenFlow</td>
</tr>
<tr>
<td>Flow Aggregation</td>
<td>Flow-1 (pkt) - Flow-3 (e.g. wave) - Flow-2 (pkt)</td>
<td>Flow modification message is to add flows to an Ethernet port, not to another flow</td>
<td>Introduce the concept of virtual port for non-packet flows</td>
</tr>
<tr>
<td>QoS and CAC</td>
<td>Flow-1 (7G) - Flow-3 (10G) - Flow-2 (3G)</td>
<td>QoS is not in v1.0</td>
<td>Largely solved in v1.3. But need to include latency requirements</td>
</tr>
<tr>
<td>Implicit provisioning</td>
<td>F1 ⊗ F2 ⊗ F3 ⊗ F4</td>
<td>How can we make flow id’s unique?</td>
<td>MPLS Label? ☺ Push this to OTS-Control to handle</td>
</tr>
</tbody>
</table>
Example: Multi-Layer Integration

- Operators may perform TE using the topology information gathered from the network
- Use simple tunneling mechanism for traffic aggregation
- Transport equipment is responsible for interworking at data-plane
Work Items

• Architecture and framework

• OpenFlow enhancement discussion, modification and prototyping

• Protocol extensions for control and management
Final Words

- SDN can speed up service deployment in transport network through network overlay and virtualization

- Open Transport Switch is a virtual switch installed on transport equipment for SDN Controller to discover, monitor and provision user flows

- We’d like to work with the community to define and develop Open Transport Switch together