Requirements for Switch Fabric and Expectations for Trellis

April 20, 2018
NTT Network Service Systems Lab.
Takayoshi Hirasawa
NTT Group

NTT Group Holding Company

- NTT
- R&D

12 Laboratories 2,500 researchers

- Information Network Laboratory Group
- Network Service Systems Lab.
- Transport Network Innovation Project (Core network development team)

★ I am here

- NTT East
- NTT West

NTT East
- Regional Communications Business

NTT West
- Communications

NTT Communications
- Dimension Data
- Long distance and international communications Business

NTT DOCOMO
- Mobile Communications business

NTT DATA
- System integration

Other group companies
- Others (real estate, finance, construction power business etc.)
Requirements for current network

- NTT groups provide various services with **high scalability and high reliability**.
- These requirements have been realized in **a dedicated high-end routers**.

**Example of NTT Regional Communications Business**

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic service</strong></td>
<td>Internet(IPv4, IPv6), telephone, telecast</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td><strong>Route</strong>: Over a few hundreds of thousands routes</td>
</tr>
<tr>
<td></td>
<td><strong>Traffic</strong>: Over tens of Tbps</td>
</tr>
<tr>
<td><strong>Additional functions</strong></td>
<td>PPPoE, IPv6 native etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reliability</strong></td>
<td>Redundancy of each function</td>
</tr>
<tr>
<td></td>
<td>Rapid switching for link failure</td>
</tr>
<tr>
<td><strong>Recovery operation</strong></td>
<td>Internet: within 2 hours</td>
</tr>
<tr>
<td></td>
<td>VoIP: within less time than internet</td>
</tr>
</tbody>
</table>
Disaggregation of high-end routers

- On the other hands, especially in OTT, a merchant silicon-based switch has great demands.
- CAPEX/OPEX savings and flexibility can be expected for a carrier with commodity products.

**Current**

- High-end core routers
  - Management
  - Routing
  - Forwarding

**Future**

- Merchant silicon-based switch clusters
  - IA server
  - Management
  - Routing
  - Forwarding

Disaggregation of the unified functions into commodity products
Activities

Our main activity
✓ Promotion of network development using commodity products

✓ Discussion with an open community about carrier requirements

Our working about ONOS/CORD
• ONOS/CORD verification (especially Trellis)
• Trying to close the gap between the results and our requirements

Today’s topic
1. The verification results of the CORD-3.0 physical PoD
2. Proposal of new Trellis architecture for improvement of reliability
1. CORD physical PoD verification
Existing network architecture

- There are two architectures regarding the deployment of the routing function.
  - Autonomous control architecture – Deploy routing functions on each switch
  - Centralized control architecture – Deploy routing functions on central controller

#### Autonomus control architecture

- **NOS control**
- Detailed components: Management
- Interfaces: NETCONF etc.

#### Centralized control architecture

- **Routing/Forwarding control**
- Detailed components: Management, Routing
- Interfaces: OpenFlow

---

**Advantage**
- Autonomous control => **Conventional C-plane processing reliability.**
- Centralized control => **Handling multiple switches as a single logical node.**

---

* NOS: Network operation system

---
1. CORD physical PoD verification

Verification condition (1/2)

**Period**
- ✓ Installation – 2017/6 - 2017/7
- ✓ Verification – 2017/8 - 2017/12

**Verification points**

<table>
<thead>
<tr>
<th>Component</th>
<th>Router</th>
<th>Autonomous MSF (NTT Lab.)</th>
<th>Centralized Trellis/CORD (ONF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Dedicated vendor Hardware</td>
<td>Commodity multivendor products</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Dedicated vendor NOS</td>
<td>Multivendor NOS + Open NOS</td>
<td>Open NOS</td>
</tr>
<tr>
<td>Internal IF</td>
<td>Dedicated vendor IF</td>
<td>Standard (Routing protocols)</td>
<td>Standard (OpenFlow)</td>
</tr>
<tr>
<td>Redundancy of processor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Routing control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Route update link</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compatibility</td>
<td>A single logical node</td>
<td>-</td>
<td>N/A</td>
</tr>
<tr>
<td>Functionality</td>
<td>L2 switching</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>IP routing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>VPN</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Maintenance</td>
<td>In-service operation</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scalability</td>
<td>Route</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Traffic</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Verification target**
(especially compatibility)
1. CORD physical PoD verification

Verification condition (2/2)

Components
✓ 5 switches (2 Leaf and 3 Spine) and 4 servers

Profile
✓ A residential profile, R-CORD (Each profile applications were not main target.)
1. CORD physical PoD verification

Summary of results

- We confirmed basic operation, but there were some lack of function.
- Especially, from viewpoint of reliability, we confirmed it indispensable to improve network architecture.

<table>
<thead>
<tr>
<th>Component</th>
<th>Router</th>
<th>Autonomous MSF (NTT Lab.)</th>
<th>Centralized Trellis/CORD (ONF)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dedicated vendor Hardware</td>
<td>Commodity multivendor products</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Dedicated vendor NOS</td>
<td>Multivendor NOS + Open NOS</td>
<td>Open NOS</td>
</tr>
<tr>
<td>Software</td>
<td>Dedicated vendor IF</td>
<td>Standard (Routing protocols)</td>
<td>Standard (OpenFlow)</td>
</tr>
<tr>
<td>Internal IF</td>
<td>Dedicated vendor IF</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Redundancy of processor</td>
<td>Forwarding control</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Routing control</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Route update link</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Compatibility</td>
<td>A single logical node</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Functionality</td>
<td>L2 switching</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>IP routing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>VPN</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Maintenance</td>
<td>In-service operation</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Scalability</td>
<td>Route</td>
<td>✓</td>
<td>Not enough</td>
</tr>
<tr>
<td></td>
<td>Traffic</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
1. CORD physical PoD verification
CORD PoD reliability

- Some essential functions are through the management switch.
  - ARP request/reply, Flow rule update, ONOS clustering connection etc.
- When the management switch failed, all connections were disconnected.

When management switch failed, both of overlay by VXLAN and underlay becomes unconnectable. It means PoD can’t provide service anymore.

It is difficult to make management switch redundant because a white-box switch has generally only one management port.
2. New architecture for Trellis
Improvement of reliability

Characteristic of centralized control
- Advantages of compatibility with existing network design.
- Disadvantages of reliability at single point of failure on the management switch.

Current architecture

We first challenged to improve reliability with keeping the advantage of the current architecture.
2. New architecture for Trellis
Solutions for improvement

1. Expansion of management network
   • An increase of management port is difficult to control by us. (It is hardware requirement)
   • Management network design becomes complicated.

2. Implementation of back up routing function to switch
   • An architecture changes from centralized to autonomous.
   • Changing the NW design will affect allover network.

3. Distribution of centralized control via in-bound connection
   • There are no influence outside PoD
   • There are no additional hardware requirements
2. New architecture for Trellis
Proposed architecture overview

- Eliminate the management switch that becomes SPOF and connects it via D-plane port.
- We considered following implementation on a switch.

**Implementation on a switch**

<table>
<thead>
<tr>
<th>Base function</th>
<th>Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Connect management port to D-plane port</td>
<td>D. Switching management connection</td>
</tr>
<tr>
<td>B. Pick up OF channel from D-plane port</td>
<td></td>
</tr>
<tr>
<td>C. Establish OF channel and update flow rules</td>
<td></td>
</tr>
</tbody>
</table>

**Verification point**

<table>
<thead>
<tr>
<th>Basic function</th>
<th>Failure scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection establishment</td>
<td>ONOS controller failure</td>
</tr>
<tr>
<td>Packet forwarding</td>
<td>Internal link failure</td>
</tr>
<tr>
<td>Flow rule update</td>
<td>Switch node failure</td>
</tr>
</tbody>
</table>

- Mgmt port
- OpenFlow channel
- D-plane port
- ONOS clustering
- Additional function
- Physical wiring

Step 1
A, B, C

Step 2
D
2. New architecture for Trellis
Initial study (Step1)

- We conducted initial study of in-bound packet control using OSS.

- Component
  - Two white-box switches (AS5812-54X)
  - Additional software in external VM (It will be put on switch in future.)
    - OpenvSwitch => for abstraction of indigo agent and management port.
    - Ryu framework => for control OF channel on D-plane

![Diagram of the network setup](image)

Initial scenario:
A sends packets to B

In an ideal model, deploy it on the switch.
2. New architecture for Trellis
Detailed packet processing (1/2)

- Role of each component
  - OVS1: Abstraction of indigo agent and storing flow rules from ONOS controller
  - OVS2: Connection between virtual interfaces and D-plane port on ASIC
  - Ryu: Control of port mapping between OVS2 and ASIC
    - Sync of flow rules between OVS1 and ASIC

1. Ryu picks up an unknown packet to mapped port.
2. OVS1 ask ONOS on VM2 about unknown packet processing via port-M on VM1
3. Ryu maps port-M and D-plane port
4. Pick up packets to itself by flow rules (MAC/IP of port-M on VM1)
2. New architecture for Trellis
Detailed packet processing (2/2)

- Role of each component
  - OVS1: Abstraction of indigo agent and storing flow rules from ONOS controller
  - OVS2: Connection between virtual interfaces and D-plane port on ASIC
  - Ryu: Control of port mapping between OVS2 and ASIC
    : Sync of flow rules between OVS1 and ASIC

---

- **External VM 1**
  - OVS1: Indigo abstraction
  - OVS2: M/D-plane integration

- **External VM 2**
  - OVS1: Indigo abstraction
  - OVS2: M/D-plane integration

---

1. Pick up packets to itself by flow rules (MAC/IP of port-M on VM1)
2. Ryu maps port-M and D-plane port
3. ONOS send new flow rules to OVS1 on VM1
4. Ryu copies flow rules from OVS1 to ASIC
5. Sync of flow rules between OVS1 and ASIC
6. Ryu maps port-M and D-plane port
7. Pick up packets to itself by flow rules (MAC/IP of port-M on VM1)
8. OVS1 stores flow rules
9. Ryu copies flow rules from OVS1 to ASIC

---

OF channel about ONOS
OF channel about Ryu
Data packet

---

Copyright©2018 NTT corp. All Rights Reserved.
2. New architecture for Trellis
Packet walkthrough

- After flow update, wideband transmission by hardware processing can be performed without any software processing.

![Diagram showing packet walkthrough]

- OF channel about ONOS
- OF channel about Ryu
- Data packet

1. Icon of detailed packet processing steps
2. New architecture for Trellis
Verification and results

- We experimentally demonstrated basic operation in two pattern.
  - Test item 1 – OF channel connection via D-plane port
  - Test item 2 – ONOS clustering connection via D-plane port

- Test item 1 – OF channel
- Test item 2 – ONOS clustering

**Verification point**

<table>
<thead>
<tr>
<th></th>
<th>Test item 1</th>
<th>Test item 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic function</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Connection estab.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Packet forwarding</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Flow rule update</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Failure scenario</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Successfully completed!
2. New architecture for Trellis

Next steps

- Implementation of Step 2 (Switching management connection)

- Expansion to CLOS topology (undergoing)

- Verification of failure scenario
  - ONOS controller failure (done)
  - Internal link failure
  - Switch node failure

- Further study
  - Add L2/L3 VPN function
  - Redundancy of routing function (Quagga)
## Expectations for Trellis

- We expect Trellis to expand further functions and the scale in the future.

### Table: Comparison of Router and Centralized Trellis/CORD (ONF)

<table>
<thead>
<tr>
<th></th>
<th>Router</th>
<th>Centralized Trellis/CORD (ONF)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redundancy of processor</strong></td>
<td>Route update link</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Forwarding control</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Routing control</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>A single logical node</td>
<td>-</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>L2 switching</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>IP routing</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>VPN</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Maintenance</strong></td>
<td>In-service operation</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Route</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Traffic</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Architecture design
- **RIB/FIB control software**
  - FRR (Quagga) enhancement
  - ONOS enhancement

### Handling ASIC
- OF-DPA
- Native SDK
- P4 (Stratum)

*There is no redundant technique to keep a single logical node not.*
For a high-reliable network

- In order to realize the high-reliable network without single point of failure, we believe that **some functions should be implemented on a switch device**.
  - Conventional architecture
    - Autonomous D-plane control by network OS
  - The proposed architecture
    - Autonomous OpenFlow/ONOS clustering control on D-plane by mini controller

- We believe it is important to utilize lightweight platform that can run on a switches. That’s why we applied Ryu framework this time.
Expectation for Stratum

- We’re also interested in Stratum to realize a widely adaptable network.

Stratum will be designed to support a variety of use cases including:

**Thin model**
for more flexible D-plane control
- Build forwarding table directly
- C/D separated architecture

**Thick model**
for more an autonomous network
- Autonomous route construction
- Distributed processing

Focusing on both models, we will continue to research a better network
Thank you for listening

We are looking for collaborators who can discuss such an approach. Please contact us if you are interested.
Reference: Multi Service Fabric

◆ An activity of our carrier-grade SDN architecture.

MSF is the architectural design of carrier-grade SDN that provides wide logical network slices with commodity products including a white-box switch.

Newly updated on 4/16 !!!


Supported functions
- Controller-independent autonomous control architecture
- L2(VXLAN)/L3(MPLS) VPN control
- Multi clusters control
- In-service operation
- High-reliability (There is no SPOF to stop services.)
- Switch set-up automation
  etc.
Reference: NTT’s activity about network OS

◆ An activity of our open network OS

The prototype of carrier-grade NOS based on open technology (OSS/Open API) with IP/MPLS capability

[Legend]

Existing Software (OSS)  Developed (OSS)