Distributed P2P ONOS Provider for a Multi-Domain Environment

Matteo Gerola, CREATE-NET
Short History

• Follow up of the ”Inter Cluster Onos Network Application (ICONA)” effort
  – GÉANT open-call project (DREAMER) focusing on a geographically distributed SDN architecture

• Open-source effort as part of the ONOS community (3 companies, 6 developers)
Problem Statement

• SDN open source initiatives tackling very useful use cases
• But... What if I have two (or n) SDN networks?
  – They can communicate through “legacy” networks
• But... Is this the best solution?
  – A lot of effort to have smart and programmable networks, but no control end-to-end
• Can I share some (minimal) information between networks?
• Can I have multi-networks intents?
Use Case: Federation of SDN NOSes

- **Service Subscriber**
  - Mobile Terminals
  - SOHO
  - Branch office
  - Head Quaters

- **Access and Aggregation**
  - Cell sites Radio Access
  - Central Office
  - Wireline Access
  - Switching Aggregation

- **Edge**
  - Metro Regional POPs
  - Edge Services Core Routing
  - Transport Switching
  - Mobile Switching
  - Mobile Aggregation and Switching

- **Core**
  - Core POPs
  - Core Routing
  - Optical Switching

- **Data Center**
  - Cloud Data Center
  - App Server and Storage
  - Edge Services
  - Service POPs
  - Edge Service
  - Specialized App Servers

#ONOSProject
Use Case: Content Delivery Network in a Multi-Domain scenario
The Multi-Domain ONOS Provider (MDOP) enables several ONOS clusters to share information about their networks.

The MDOP can be applied both to single and multi domain use-cases.

- The east-west interface (termed Remote Communication) is based on a peer-to-peer approach.

- The MDOP enables all ONOS applications to configure, via the ONOS APIs, L2 and L3 routes crossing different dataplane devices which may not be under the control of the specified cluster.

- A policy-based approach is considered to cover varying use-cases.
Why such a distributed architecture?

- “Legacy” internet:
  - Intra-domain
    - Interior Gateway Protocols (IGPs), route-based, like OSPF and IS-IS
  - Inter-domains
    - Exterior Gateway Protocols (EGPs), policy based (BGP)

- Different requirements bring to various protocols and solutions

#ONOSProject
High Level Architecture

Cluster A

- Apps
- NB Core API
- Distributed Core
  (state management, notifications, high-availability & scale-out)
- SB Core API

Providers

Protocols

Cluster B

- Apps
- NB Core API
- Distributed Core
  (state management, notifications, high-availability & scale-out)
- SB Core API

Providers

Protocols

Multi-Clusters Peering Provider

Remote Communication

Customer Edge Point

East West interface

InterLink

#ONOSProject
Topology Abstractions
Interactions with the MDOP

• The MDOP is completely transparent to the ONOS client applications:
  – External domains are exposed to the ONOS core as single or multi devices (depending on the abstraction selected)
  – These devices are seen as “standard” ONOS switches, tagged with “external-domain” vendor field

• Apps/Users can interact to the external devices through:
  – ONOS GUI
  – ONOS CLI
  – ONOS REST
  – ONOS Java APIs (Intents, FlowObjectives, ...)
• Network managers can configure at runtime:
  – peering **clusters** allowed to access the local information
  – **maximum number of intents** settable by remote clusters
  – **priority** of the remote intents
  – **weight** of each **interlink**
  – **preferred paths** for specific classes of traffic (based on L2 and L3 fields)
  – …
Remote Communication

The communication will be established through BGP Implementation:

- BGP messages can carry capability and reachability information as part of their message format.
- BGP has inter-AS communication capability which makes it apt for any peer ONOS cluster data to be exchanged/shared.
- BGP General Approach:
  - Functional State Machine approach makes it easier to implement a BGP session.
  - BGP can be used independently as an application layer protocol (on need basis).
  - The straightforward message format is in accordance with the OpenFlow messages.
- The implementation can be flexible to enhance for extra data.
- Standardized approach as mentioned in the draft: [https://tools.ietf.org/html/draft-yin-sdn-sdni-00](https://tools.ietf.org/html/draft-yin-sdn-sdni-00)
Remote Communication Architecture

- Key Points:
  - MDOP to generate data for communication through Rest (under discussion)
  - Dynamic database to maintain BGP information (Sqlite3)
  - Use existing ONOS-BGP or configure Quagga for the BGP session (under discussion)
## Information Shared between domains

<table>
<thead>
<tr>
<th>Topology Information</th>
<th>TE information</th>
<th>Flow specification</th>
<th>Extendable to</th>
</tr>
</thead>
<tbody>
<tr>
<td>• L0 INFO</td>
<td>• Master controller IP address (optional)</td>
<td>• Create-Read-Update-Delete FlowRule</td>
<td>• Network events</td>
</tr>
<tr>
<td>SwitchId</td>
<td>• No of hops</td>
<td>• FlowRule Serialization</td>
<td>• User defined request information</td>
</tr>
<tr>
<td>PortId</td>
<td>• Aggregated latency</td>
<td>• Traffic Selector (match fields)</td>
<td>• QoS/Topology requirements from user application request</td>
</tr>
<tr>
<td>• L2 INFO</td>
<td>• Bottleneck bandwidth</td>
<td>• Traffic Treatment (actions)</td>
<td></td>
</tr>
<tr>
<td>L2 Address</td>
<td>• Reserved bandwidth</td>
<td>• Flow Priority</td>
<td></td>
</tr>
<tr>
<td>L2 VLAN</td>
<td>• Estimated current load</td>
<td>• Preferred path on QoS approximation (optional)</td>
<td></td>
</tr>
<tr>
<td>• L3 INFO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3 Addresses</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3 subnets</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Software Defined Transformation of Service Provider Networks

Q&A

Join the journey @ onosproject.org

#ONOSProject