AutoVFlow: Autonomous Virtualization for Wide-area OpenFlow Networks

Hiroaki Yamanaka, Eiji Kawai, Shuji Ishii, and Shinji Shimojo (NICT)
@EWSDN 2014
3 SEP 2014
Agenda

• Background and Problem
• Proposal of AutoVFlow
• Evaluation
• Conclusion and Future Work
Background and Problem
Background

• Various clean-slate network designs for various applications
  – Customized protocol for each application
    • Named data networks, MobilityFirst Project, etc.
  – Using SDN techniques (e.g., OpenFlow): an easy way to realize clean-slate network design

• Demand for deploying clean-slate network designs on wide-area networks
  – Many Internet-wide applications
Multi-tenancy in OpenFlow

- Building multiple virtual OpenFlow networks on wide-area networks
  - Easy to implement various clean-slate network designs on shared infrastructures
  - Improving the cost-effectiveness of wide-area OpenFlow network infrastructures
- We need to consider building virtual OpenFlow networks on multiple administration networks.
Flow Space Virtualization

- OpenVirteX [1] virtualizes the flow space (i.e., header values)
  - Enabling to use arbitrary (even, overlapped) header values in all virtual OpenFlow networks
- Accommodating many tenants easily
  - There is no need to agree the flow space division among tenants.
- Full-programmability of OpenFlow for tenants
  - Arbitrary header values can be used in virtual OpenFlow networks

![Diagram showing the Flow Space Virtualization process with virtual OpenFlow networks, OpenVirteX proxy, and substrate OpenFlow network.]

Virtualization using Single-proxy Architecture in a Wide-area Network

- A single proxy is enforced to have **too much responsibility** in wide-area networks
  - When the proxy is failure, it affects all virtual networks provided by all administrators.
Virtualization over Multiple Administration Networks

- **AutoSlice [2]**
  - Building virtual OpenFlow networks over multiple administration networks
    - Multi-proxy architecture **improving robustness** of the system
  - VLAN-based flow space virtualization
    - A single third party solely manages the allocations of VLAN IDs to the tenants.
    - Tenants cannot control VLAN tag.

![Diagram of virtualization](image)

**Table:**

<table>
<thead>
<tr>
<th>Tenant ID</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>110</td>
</tr>
<tr>
<td>β</td>
<td>120</td>
</tr>
</tbody>
</table>

Flow Space Virtualization using Multi-proxy Architecture

• If flow space virtualization is implemented using the multi-proxy architecture, a single third party still has large responsibility.

Every time when the new header value is used, the mapping of header values is updated.
Problem on Flow Space Virtualization in a Wide-area Network

• Flow space virtualization is necessary for various clean-slate network designs on wide-area networks.
  – Proposed by OpenVirteX
    • scaling-up multi-tenancy
    • supporting full-programmability of OpenFlow
  – Different from FlowVisor
  – Not supported in AutoSlice

• Problem
  – Using existing techniques, a single party is enforced to have large responsibility to implement flow space virtualization in a wide-area network.
Proposal of AutoVFlow
Summary of AutoVFlow

- **AutoVFlow**
  - No need of a third party to implement flow space virtualization
  - The proxies of substrate network administrators autonomously translate header value.
  - A failure of an administrator affect only the part of virtual OpenFlow switches.
Translation for Own Physical OpenFlow Switches

• Translation in each substrate OpenFlow network is based on the local mapping table.

<table>
<thead>
<tr>
<th>Tenant ID</th>
<th>Virtual MAC address</th>
<th>Physical MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>h</td>
<td>h&lt;sub&gt;α&lt;/sub&gt;</td>
</tr>
<tr>
<td>β</td>
<td>h</td>
<td>h&lt;sub&gt;β&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

Matching rule: h<sub>α</sub>
Matching rule: h<sub>β</sub>
Matching rule: h

Translating header values in OpenFlow protocol messages

Modify or recovery end-host’s data packet header
Translation for Data Packets Transferred to Adjacent Substrate Network

- **Sender administrator proxy**
  - Setting the physical header values of the data packets to the ones in the receiver substrate network before sending

- **Receiver administrator proxy**
  - Handling data packets using local mapping table

![Diagram showing data packets and header values](image)

- AutoVFlow proxy A (sender)
- AutoVFlow proxy B (receiver)

Q: What does \( h \) in \( \alpha \) correspond in your substrate network?

<table>
<thead>
<tr>
<th>Tenant ID</th>
<th>Virtual MAC address</th>
<th>Physical MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>( h )</td>
<td>( h_{B\alpha} )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>( h )</td>
<td>( h_{B\beta} )</td>
</tr>
</tbody>
</table>

Flow-mod (match \( h \))

Flow-mod (rewrite to \( h_{B\alpha} \))

Tenant ID:
- \( \alpha \)
- \( \beta \)

Virtual OpenFlow networks:
- \( \alpha \)
- \( \beta \)

Flow-mod tables:
- \( A: h_{B\alpha} \)

Local mapping table B:

Administrator A

Administrator B

Virtual MAC address

Physical MAC address
Evaluation
Evaluation

- Objective: evaluating the overhead on the data packet transmission performance
  - For translation, data packet headers can be modified at high-speed at ingress and egress hardware switches in a data plane.
    - However, AutoVFlow proxies affect communications in a control plane.
- Experiments: measuring the delay in OpenFlow protocol message communications for handling data packet transmission
  - Scenario: reactive flow setting
  - Delay of the messages affects the data packet transmission performance.
Delay in packet-in and packet-out message communications

- Packet-in message
- About 1 ms overhead in average
- Negligible overhead: there is 1 ms delay even in the case of “without AutoVFlow”.

- Packet-out message
Measuring the delay in flow-mod message communications

- **The worst case scenario** in AutoVFlow for flow-mod message
  - Involving two AutoVFlow proxies
Delay in flow-mod message communications

- 5.85 ms overhead in average
- This overhead for flow-entry setting is negligible, compared to the greater delay in end-to-end data packet transmission in wide-area networks.
Conclusion and Future Work
Conclusion and Future Work

• Conclusion
  – The AutoVFlow mechanism allows wide-area virtual OpenFlow networks with flow space virtualization.
  – Any single party does not have large responsibility.
  – The overhead is negligible for data packet transmission performance in wide-area networks.

• Future work
  – Handling massive number of substrate OpenFlow switches
    • Improving the scalability in terms of the number of substrate OpenFlow switches within a single administration network.