IO Visor-based Packet Tracing and Collection over Distributed SmartX Server-Switch Boxes

Jungi Lee, Taekho Nam,
Aris Cahyadi Risdianto, and JongWon Kim
thnam@nm.gist.ac.kr

Networked Computing Systems Lab.,
School of Electrical Engineering and Computer Science,
Gwangju Institute of Science and Technology (GIST)
• Introduction: Background and Motivation

• Playground with Distributed Cloud-ready Boxes

• Layer 2 and Layer 3 Inter-Connections for distributed Cloud-ready Boxes

• IO Visor-based Packet Tracing and Collection

• Conclusion
Introduction: Background and Motivation

• Multiple cloud-ready SmartX Server-Switch boxes (i.e., SmartX Box Type S#) are deployed over KOREN (Korea Advanced Research Network).

• How to inter-connect distributed resource boxes to provide an environment for running application and/or services?

• How to provide flexible provisioning of L2/L3 Inter-connection between boxes?

• How to control and secure the distributed resource boxes at the center?
Playground with Distributed SmartX Server-Switch Boxes

- Znyx B1 Server-Switch: Specification

<table>
<thead>
<tr>
<th>Switch Environment</th>
<th>Znyx B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Family</td>
<td>Intel® Xeon® E5-2600v2</td>
</tr>
<tr>
<td>Switch Fabric</td>
<td>Up to 480Gbps throughput</td>
</tr>
<tr>
<td>External Interfaces</td>
<td>(24) 1G/10G supporting copper and fiber (SFP/SFP+)</td>
</tr>
<tr>
<td>Software Support</td>
<td>OpenArchitect® 4.0 - Ubuntu LTS with KVM</td>
</tr>
<tr>
<td>Protocol Support</td>
<td>Quagga L3 Protocol Suite &amp; etc…</td>
</tr>
</tbody>
</table>

http://www.znyx.com/products/hardware/b1-specifications/
Layer 2 and Layer 3 Inter-Connections for distributed Cloud-ready Boxes

<table>
<thead>
<tr>
<th></th>
<th>How to find the path</th>
<th>How to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2 (VXLAN)</td>
<td>Multi-cast between VTEPs (Vxlan Tunnel End Points)</td>
<td>Connect VM/Container with vSwitch</td>
</tr>
<tr>
<td>L3 (BGP)</td>
<td>BGP routing</td>
<td>Connect VM/Container with vRouter</td>
</tr>
</tbody>
</table>

- In case of L2 inter-connection, VXLAN tunneling can be used to implement inter-connection between boxes in the form of overlay networking regardless of the actual physical network environment.

- In the case of L3 inter-connect, virtual router and virtual switch are used inside the cloud-ready box to support L3 routing of nodes inside the box.
L3 Inter-Connection: Design and Implementation inside Server-switch Box
Flexible L2/L3 Inter-Connections Over Distributed Server-Switch Boxes

- VxLan allows L2 connections between nodes with overlay network
- BGP L3 connection allows BGP routing between nodes
Preliminary Implementation and Verifications

- Network Topology
- L3 Routing rules
- The port information inside each box.
• IO Visor?

• IO Visor is an open-source collaborative project designed to accelerate the innovation, development and sharing of virtualized kernel I/O services for many networking-related functions.

• IO Visor can be effectively exploited in many areas that include networking, security, and tracing. Specifically for packet tracing functionality, it utilize a BCC (BPF Compiler Collection) to implement IO Visor-based I/O-level packet tracing.
Design: IO Visor-based Packet Tracing and Collection

User space

Kernel space

IO Visor based Packet-precise Tracing

IP version, source IP address, destination IP address, destination port

Flows

Box

Box
Implementation and Verifications of IO Visor-based Packet Tracing/Collection
Conclusion: IO Visor-based Packet Tracing/Collection over Distributed SmartX Server-Switch Boxes.

Aspect of Operator
Provides secured operation of playground by using information of malicious packets received from 3rd-party monitoring post and ONOS SDN Controller.

Packet Tracing/Collection with IO Visor

L2/L3 Inter-Connection

Aspect of User
When running applications or services in the playground, users can take advantage of flexible L2/L3-based connections between nodes in a secured playground.
Future Works: IO Visor for Site Visibility Framework

By extending the basic functionality of IO Visor-based packet tracing and collection over distributed server-switch boxes,

We re-design and develop a prototype-level site visibility framework with DevOps concept to inspect all packets passing through multiple network interfaces at the same time.

Site visibility framework is Django-based software framework that leveraging IO Visor-based packet tracing and collection.

It can support the visibility visualization of traced packets and provide associated APIs.
Thank You