Network-virtualization Nodes that Support Mutually Independent Development and Evolution of Node Components

Yasusi Kanada & Kei Shiraishi, Hitachi Ltd.
Akihiro Nakao, University of Tokyo
Introduction: Previous Work

► We are developing VNode — deeply-programmable physical node for network-virtualization platforms — in a collaborative project.

◆ Deeply-programmable: packet data processing, such as new L3 protocol processing, can be programmable.

► A VNode consists of two types of components.
  ◆ Deeply-programmable computational component called “programmer”.
  ◆ Networking component called “redirector”.

► Programmer and Redirector are developed and evolve independently in the VNode architecture proposed by Nakao [Nak 12b].
Introduction: Today’s Topic

Redirector plays the central role in implementing two functions of VNodes, which enables the independence.

◆ Creation of external and internal model-implementation mappings:
  ● Mapping between virtual links to external physical paths
  ● Mapping between virtual links to internal physical paths of VNodes

◆ High-performance data conversion, which connects the external and internal data formats.
Network Virtualization Architecture and Platform

► Network Virtualization Architecture and Platform are developed in the collaborative project.
► Multiple slices can be created on a physical network in this architecture.
  ◆ Slices means virtual networks.

Slice developer
Slice definition

<Vnode ...

Domain Controller
Virtualization platform

Slice 1
Slice 2
Slice 3
VNode

VNode (virtualization node) is a component of the network virtualization platform.

- VNode is a physical node.
- VNode forwards packets on the platform as a router.
- Slices are implemented as overlay networks on the virtualization platform.
- VNodes are connected by tunnels using GRE/IP.
  - GRE (Generic Routing Encapsulation) is a protocol standardized by IETF.
Components of VNode

► Programmer
◆ is a programmable component that processes packets on the slices.

► Redirector
◆ forwards (redirects) packets from another VNode to a programmer and forwards packets from a programmer to another VNode.
◆ is a component that can forward or route packets on the platform.

► VNode Manager
◆ is a software component that manages the VNode.
Independently Evolvable VNode Architecture

► Programmer and Redirector are “separated” in this architecture.

► This separation means the protocols of internal and external networks can be independently defined.
  ◆ Internal network – between Programmer and Redirector.
  ◆ External network – between VNodes.

► This separation enables various evolution of VNode.
  ◆ Various types of Programmers can be connected to a Redirector.
  ◆ Various types of Redirectors can be connected to a Programmer.

► Our challenge is to implement this architecture and to obtain high performance.
Structure of Redirector

- Management of both external and internal information
- NP board that can convert data from external to internal or vice versa

Redirector
- Redirector Manager (RM)
- Control Plane (C-Plane)
- Service Module Card (SMC)
- High-end L3 Switch (RB)
- Data Plane (D-Plane)
- Internal Data Plane
Model-Mappings and Separation

- In a VNode, the model (slice part) is mapped to the internal representation.
- Out of a VNode, the model (slice part) is mapped to the external representation.
- These mappings must be separated for the sake of independent evolution.
Management of Model Mappings

► Internal mapping is collaboratively managed by Redirector and Programmer.
► External mapping is collaboratively managed by Redirector and VNode Manager.
► Therefore, Redirector plays the central role in separating and connecting the model mappings.
Conversion between External and Internal Representations

- Redirector (SMC) converts data packets between external and internal representations.
  - The throughput is 10 Gbps (when the packet size is around 1000B).
Evaluation: Mapping creation and deletion

The creation and deletion of a slice were measured.

- **Slice structure**
  - User terminal → AGW → Link sliver → Node sliver → Link sliver → Node sliver → Link sliver → AGW → User terminal

**Performance evaluation of the internal and external mappings**
- The mappings are concurrently created by the redirector when creating link slivers.

<table>
<thead>
<tr>
<th>VNode parts</th>
<th>Time for “run” (s)</th>
<th>Time for “shutdown” (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Std dev</td>
</tr>
<tr>
<td>Redirector</td>
<td>23.4</td>
<td>1.6</td>
</tr>
<tr>
<td>Programmer</td>
<td>50.3</td>
<td>3.0</td>
</tr>
<tr>
<td>Whole VNode</td>
<td>80.5</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Reserve slice       | Run slice       | Internal and external mappings of three link slivers are created by two redirectors in two VNodes
Redirectors         | Programmers     | Whole VNodes

\[-7.2\] \[0\] \[23.4\] \[50.3\] \[80.5\] Time (s)
Observation

- The overhead caused by link creation is mostly hidden if the number of virtual links per virtual node is five or less (i.e., in normal conditions).
- The overhead can still be seen in link deletion because programmers reply to “shutdown” command immediately, but it is less critical.
  - It is less critical because new links can be created while deleting old links.
Evaluation: Data Conversion

► Data conversion rate is 5-Gbps at maximum when input and output data rates of VNodes are equal.
  ◆ VNodes are connected by 10-Gbps physical links.
  ◆ The data conversion is performed on the SMC.
  ◆ Each VNode has only one SMC that is used for both direction (internal ⇔ external).

► In future, this performance can be improved by adding SMCs.
Conclusion

► To enable independent evolution of Programmer and Redirector, Redirector is designed ...
  ◆ Redirector creates two separate mappings between virtual links to external and internal physical-paths.
  ◆ Redirector implements conversion between the external and internal data formats using SMCs.

► Performance of mappings and data conversion are evaluated, and Redirector mostly satisfies the requirements.
  ◆ Time for virtual-link creation is usually shorter than other set-up times.
  ◆ The data conversion can be performed at a rate of 5 Gbps.

► Future work
  ◆ To enhance mapping and data-conversion methods for various programmers and external networks and for performance improvements.
  ◆ To reduce time for virtual-link deletion.