Thinking about Extensibility and Scalability in OpenFlow Networking
(CANS 2011)

Zhiliang WANG, Xia YIN, Jun WANG
Tsinghua University
2011-8-9
Outline

• OpenFlow Introduction

• Our ideas on Openflow Research
  – Extensibility
  – Scalability

• Summary
OpenFlow Switching

OpenFlow Switch specification

OpenFlow Switch

Secure Channel

Flow Table

Controller

PC

From: Nick McKeown’s OpenFlow Presentation
Flow Table (Spec v1.1.0)

Match Fields ➔ Counter ➔ Instructions

Match Fields:
- Switch Port
- MAC src
- MAC dst
- Eth type
- VLAN ID
- IPv4 Src
- IPv4 Dst
- IPv4 Prot
- TCP sport
- TCP dport

Counter:
- Packets
- Statistics

Instructions:
- To change action sets:
  - Forward packet to port(s)
  - Drop packet
  - Send to normal processing pipeline
  - Encapsulate and forward to controller
OpenFlow is …

• Experimental platform
  – Run experiments in production networks
  – “Enabling innovation on campus”

• New network architecture
  – Open Programmable Networking
  – Software Defined Networking
SDN: Software Defined Networking

1. Open interface to hardware
   - OpenFlow

2. At least one Network OS probably many.
   - Open- and closed-source

3. Well-defined open API

From: Nick McKeown’s SDN Presentation
ONF: Open Networking Foundation

• a nonprofit organization dedicated to promoting new approach of Software-Defined Networking (SDN)
• Founded in March, 2011
• Standardization of OpenFlow Specification
• Member including:
  – ISP, ICP, equipment vendors, networking and virtualization software suppliers, and chip technology providers
Our ideas on OpenFlow Research

Scalability
Extensibility
IPv6 Extensions

From: OpenFlow Specification v1.1.0
IPv6 Extensions of OpenFlow
IPv6 Extensions of OpenFlow

• Why OpenFlow with IPv6 support?

• The design principle of IPv6 extension

• Next steps
Why OpenFlow with IPv6 support?
Design Principle

• At least two data structures in OpenFlow needed to extend

- Flow Table
- Group Table

• Additional instructions or actions needed to design to support the new data structures
Flow Table

- Extending the Flow Table to support IPv6
Flow Table

- Flow Table supporting IPv4/IPv6

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>...</th>
<th>IPv4 Src</th>
<th>IPv4 Dst</th>
<th>IPv4 Prot</th>
<th>Tran sport/ICMP Type</th>
<th>Trans dport/ICMP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>32bits</td>
<td>32bits</td>
<td>8bits</td>
<td>16bits</td>
<td>16bits</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>...</th>
<th>IPv4 Src/IPv6 Src</th>
<th>IPv4 Dst/IPv6 Dst</th>
<th>IPv4 Prot/IPv6 Nxt Hdr</th>
<th>Tran sport/ICMPv4 Type</th>
<th>Tran sport/ICMPv6 Type</th>
<th>Trans dport/ICMPv4 Code/ICMPv6 Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>128bits</td>
<td>128bits</td>
<td>8bits</td>
<td>16bits</td>
<td>16bits</td>
<td>16bits</td>
</tr>
</tbody>
</table>
Group Table

- Defined in OpenFlow version 1.1
- Multicast/Broadcast/Anycast can be supported by different group type.

IPv6 Dst == Multicast Addr.  Action: Group 100

<table>
<thead>
<tr>
<th>Match Fields</th>
<th>Counter</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 Dst</td>
<td>Port 1:</td>
<td>Group 100</td>
</tr>
<tr>
<td>Port 3:</td>
<td>output</td>
<td></td>
</tr>
<tr>
<td>Port 4:</td>
<td>output</td>
<td></td>
</tr>
<tr>
<td>Port 5:</td>
<td>output</td>
<td></td>
</tr>
</tbody>
</table>

Port 1: output
Port 3: output
Port 4: output
......
Group Table

- IPv6 has well-known multicast addresses
  - all-nodes multicast address(FF01::1)
  - all-router multicast address(FF01::2)
  - .......
  - These well-known multicast addresses should have predefined group identifier.
  - The corresponding action buckets should also be assigned by the controller as one command.
Next Steps:

• Submit the proposal of OpenFlow IPv6 extensions to ONF

• Prompt standardization of OpenFlow with IPv6 supports

• Demos of OpenFlow IPv6 extensions
Scalability in OpenFlow Networking
Scalability in OpenFlow Networking

• Scalability issues

• Related work

• Our ideas

• More Further…
Scalability Issues

• **Centralized** OpenFlow Control Plane is lack of scalability

• Problems
  – There is only one single controller in current deployments
    • Performance issues of controller in large scale network
  – Interconnections of remote OpenFlow Islands
    • Large establishment delay of new flow entry
Related Work to Solve Scalability Issues

• 1. Using optimization techniques
  – Maestro uses the technique of parallelism
    • multi-threading to handle the flow requests from OpenFlow switches
    • batching sending when the controller sends control configurations back to the switches

Related Work to Solve Scalability Issues

• 2. Devolving some control functions back to the switches
  – DevoFlow (Devolved OpenFlow) [HotNets10][Sigcomm11]
    • Decreasing the interactions between OpenFlow switches and controller
  – DIFANE (DIstributed Flow Architecture for Networked Enterprises) [Sigcomm10]
    • Distributing the rules across “authority switches”

Related Work to Solve Scalability Issues

• 3. Designing a distributed control platform
  – **HyperFlow** [INM/WREN 10]
    • Distributed event-based control plane for OpenFlow,
    • logically centralized but *physically distributed*
  – **Onix** [OSDI10]
    • Distributed control platform for large-scale networks
    • providing more general APIs than previous systems and flexible distribution primitives

Our Ideas

• Scalable Distributed Control Platform
  – physically distributed
  – logically distributed
  – Each controller serves for its managed OpenFlow Cloud
  – Control Platform Overlay
  – Control Protocol
More Further…

• **Inter-Domain Scenarios?**
Summary

• Extensibility of OpenFlow
  – IPv6 Extensions

• Scalability of OpenFlow
  – Scalable Distributed Control Platform
Thank you!

Q&A

http://netarchlab.tsinghua.edu.cn/~wzl
wzl@cernet.edu.cn