OpenFlow MPLS and the Open Source Label Switched Router

James Kempf*, Scott Whyte†, Jonathan Ellithorpe‡, Peyman Kazemian‡,
Mart Haitjema**, Neda Beheshti*,
Stephan Stuart†, Howard Green*
Outline

• OpenFlow Architecture
• OpenFlow MPLS Design
• NetFPGA Implementation
• The Open Source LSR
• Summary
OpenFlow Architecture
OpenFlow System Architecture

Note: In the Open Source LSR, the controller and switch are in the same chassis!
OpenFlow Switch Architecture

OpenFlow Flow Table Entry

<table>
<thead>
<tr>
<th>Rule</th>
<th>Action</th>
<th>Stats</th>
</tr>
</thead>
</table>

- **Rule:**
  - Defines the flow
  - Currently a 10-tuple: [OFSpec]

<table>
<thead>
<tr>
<th>In Port</th>
<th>VLAN ID</th>
<th>Ethernet</th>
<th>IP</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SA</td>
<td>DA</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SA</td>
<td>DA</td>
<td>Proto</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Src</td>
<td>Dst</td>
<td></td>
</tr>
</tbody>
</table>

- **Action:**
  - Specifies what action the switch should take
  - Very simple actions such as:
    - Forward out port, send to controller, drop packet
    - Rewrite certain header fields (with fixed values)

- **Stats:**
  - Packet & Byte counters
OpenFlow MPLS Design
OpenFlow MPLS Switch
Architecture Changes

Matching
• Extend 10-tuple to include top 2 labels in label stack
• Most usage scenarios covered by 2 labels
• Label is 20-bit label (not exp, ttl, stack)

<table>
<thead>
<tr>
<th>In Port</th>
<th>VLAN ID</th>
<th>Ethernet</th>
<th>MPLS</th>
<th>IP</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SA</td>
<td>DA</td>
<td>Type</td>
<td>Label 2</td>
</tr>
</tbody>
</table>

Actions
• Send packet to virtual port to perform complex protocol specific actions (i.e. MPLS pop, push, swap)
• Drop TTL expired packets & add mpls_ttl_exp stat
Virtual Port Abstraction

- **Virtual Ports**
  - Extend notion of virtual ports
  - Virtual ports handle complex protocol/port specific actions

- **Virtual Port Table**
  - Similar to flow table
    - Virtual port number instead of 10-tuple
    - List of actions to perform
  - Parent port may be
    - physical port
    - another virtual port

<table>
<thead>
<tr>
<th>Physical Ports (0-0xff00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused (0xff01-0xffff8)</td>
</tr>
<tr>
<td>OpenFlow Virtual Ports (0xffff9-0xffffffff)</td>
</tr>
<tr>
<td>Virtual Ports (0x10000-0x40000000)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port No</th>
<th>Parent Port</th>
<th>Actions</th>
<th>Stats</th>
</tr>
</thead>
</table>

Virtual Port Table Entry
Virtual Port Actions

**Virtual Port Table Actions:**

- **`push_mpls` action**
  - Push a 32-bit MPLS label onto top of label stack
  - Copies TTL and Exp bits from IP header if required
  - Parameters:
    - 20-bit label to push
    - 8-bit ttl value (optional)
    - 3-bit exp value (optional)
    - Flags (to indicate special behavior)

- **`pop_mpls` action**
  - Pop the top 32-bit MPLS label off the label stack
  - Copies TTL and Exp bits to IP header if required
  - Parameters
    - Eth_type (if popping last label)
    - Flags (to indicate special behavior)
OpenFlow MPLS Changes

Changes to Flow Table:
- **Output action**
  - Port number now 32-bits

- **Set_mpls_label action**
  - Rewrite 20-bit label on top label
  - Similar to existing set_vlan_id action

- **Set_mpls_exp action**
  - Rewrite 3-bit exp field on top label
  - Similar to existing set_vlan_pcp action

New OpenFlow Messages
- **vport_mod message**
  - Add or remove a virtual port table entry
  - Parameters:
    - Parent port number
    - Array of virtual port actions

- **vport_table_stats message**
  - Stats for virtual port table
  - Includes
    - max_vports
    - active_vports
    - lookup_count
    - port_match_count
    - chain_match_count

- **port_stats message**
  - Now also used for virtual ports
  - Only tx_bytes and tx_packets fields are used
NetFPGA Implementation
NetFPGA Hardware

- 4 1G Ethernet ports
- Standard PCI form factor
- Xilinx Virtex-II Pro 50 FPGA
  - 53K logic cells
  - 4M block RAM
  - 700K distributed RAM
  - 2 Power PC cores
- Open source software
OpenFlow MPLS NetFPGA Implementation

- Maximum two virtual port cascades.
  - Virtual ports must be of the same type
- 64 “push” virtual ports and 64 “pop” virtual ports
- Aggregated statistics on virtual port cascades
  - No per virtual port statistics
The Open Source LSR
Interoperability Tests
Comparative Forwarding Performance

![Chart showing comparative forwarding performance between open source LSR and mpls-linux. The chart displays data for packet sizes of 68 Bytes and 1504 Bytes, with open source LSR significantly outperforming mpls-linux in terms of megabits per second.]
Summary
Summary

• Extension of OpenFlow 1.0 switch model to include MPLS
  – Two labels
  – Virtual port abstraction for MPLS header operations

• Software and hardware implementation
  – OpenVSwitch and Stanford Reference Switch software implementations
  – NetFPGA hardware implementation

• Open source Label Switched Router (LSR) implementation
  – OpenFlow controller runs on the same chassis as switch
  – Software available at http://code.google.com/p/opensource-lsr

• Test results
  – Superior performance to a software implementation
  – Interoperates with commercial IP/MPLS routers