Openflow in two hours

Indiana Center for Network Translational Research and Education

the research arm of

GlobalNOC
Agenda

Overview of OpenFlow

Configure HP switches for OpenFlow

Use Open DayLight to control switches

Watch with WireShark

Play with alternate switch topologies
OpenFlow Controller

Features

Value Add

OpenFlow Protocol
Each switch connects directly with OF Controller
# Flow Table

<table>
<thead>
<tr>
<th>Header Fields</th>
<th>Counters</th>
<th>Actions</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress Port</td>
<td>Per Flow Counters</td>
<td>Forward</td>
<td></td>
</tr>
<tr>
<td>Ethernet Source Addr</td>
<td>Received Packets</td>
<td>(All, Controller, Local, Table, IN_port, Port#)</td>
<td></td>
</tr>
<tr>
<td>Ethernet Dest Addr</td>
<td>Received Bytes</td>
<td>Normal, Flood</td>
<td></td>
</tr>
<tr>
<td>Ethernet Type</td>
<td>Duration seconds</td>
<td>Enqueue</td>
<td></td>
</tr>
<tr>
<td>VLAN id</td>
<td>Duration nanoseconds</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>VLAN Priority</td>
<td></td>
<td>Modify-Field</td>
<td></td>
</tr>
<tr>
<td>IP Source Addr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Dest Addr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Protocol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP ToS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP code</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Per Flow Counters**
- Received Packets
- Received Bytes
- Duration seconds
- Duration nanoseconds
<table>
<thead>
<tr>
<th>Header Fields</th>
<th>Counters</th>
<th>Actions</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ingress port == 2</td>
<td></td>
<td>Drop packet</td>
<td>32768</td>
</tr>
<tr>
<td>if IP_addr == 129.79.1.1</td>
<td></td>
<td>re-write to 10.0.1.1, forward port 3</td>
<td>32768</td>
</tr>
<tr>
<td>if Eth Addr == 00:45:23</td>
<td></td>
<td>add VLAN id 110, forward port 2</td>
<td>32768</td>
</tr>
<tr>
<td>if ingress port == 4</td>
<td></td>
<td>forward port 5, 6</td>
<td>32768</td>
</tr>
<tr>
<td>if Eth Type == ARP</td>
<td></td>
<td>forward CONTROLLER</td>
<td>32768</td>
</tr>
<tr>
<td>If ingress port == 2 &amp;&amp; Eth Type == ARP</td>
<td></td>
<td>forward NORMAL</td>
<td>40000</td>
</tr>
</tbody>
</table>
Special Ports

Controller (sends packet to the controller)

Normal (sends packet to non-openflow function of switch)

Local (can be used for in-band controller connection)

Flood (flood the packet using normal pipeline)
# Flow Table

<table>
<thead>
<tr>
<th>Header Fields</th>
<th>Counters</th>
<th>Actions</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ingress port == 2</td>
<td></td>
<td>Drop packet</td>
<td>32768</td>
</tr>
<tr>
<td>if IP_addr == 129.79.1.1</td>
<td></td>
<td>re-write to 10.0.1.1, forward port 3</td>
<td>32768</td>
</tr>
</tbody>
</table>

Each Flow Table entry has two timers:  

- **idle_timeout**: seconds of no matching packets after which the flow is removed. Zero means never timeout.
- **hard_timeout**: seconds after which the flow is removed. Zero means never timeout.

If both **idle_timeout** and **hard_timeout** are set, then the flow is removed when the first of the two expires.
Populating the Flow Table

Proactive
Rules are relatively static, controller places rules in switch before they are required.

Reactive
Rules are dynamic. Packets which have no match are sent to the controller (packet in). Controller creates appropriate rule and sends packet back to switch (packet out) for processing.
Controller and Switch Communication

● Mode - Controller vs. Listener
  ○ TCP Communication, who initiates conversation

● Mode and Populating Flow Table independent
Example application: topology discovery
Bootstrapping a new switch

Switch requires minimal initial configuration (e. g., IP address, default GW, and OpenFlow controller)

Switch connects to controller. Controller requests things like a list of ports, etc.

Controller proceeds to determine the switch's location.
Bootstrapping a new switch

Controller *proactively* places a rule in the switch.

If ether_type = LLDP, actions=output:controller

Then the controller creates an LLDP packet, sends it to the switch, and instructs the switch to send it out a port (repeat for all ports).

Since all switches in the controller's network have a rule to send LLDP packets to the controller, the controller is able to determine the topology.
Preparing for the hands-on stuff

Connect to the wireless OpenFlow SSID

Point your web browser to: http://172.16.1.95/smokeping/
Select “Targets” then “Pitcher”

Launch Open DayLight (run.bat or run.sh)

Launch Wireshark and configure it to monitor “openflow” traffic (may have to set decode as under “Analyze menu”

Point a second browser windows to 127.0.0.1:8080
username admin
password admin
Of Note!

Five of you will be configuring the same switch at the same time, so careful.

The vLANs are already defined. Use “write t” to get a feel for the configuration.

vLAN 1 is reserved for telnet access and switch-to-controller connections.
What the network looks like

While the vLAN remains configured for native Ethernet switching, smoke ping should indicate positive connectivity!
Today’s exercise

Enable OpenFlow on your vLAN, and point it to the controller running on your laptop.

1. Create a controller ID that points to your laptop
2. Create an OpenFlow instance for your vLAN
3. Make your vLAN a member of your OpenFlow instance
4. Point your OpenFlow instance to your controller ID
5. Enable OpenFlow for your OpenFlow instance
Create a controller ID that points to your laptop

Of Note: the switches have an OpenFlow context pre-configured

switch1# config
switch1(config)# openflow
switch1(openflow)#
controller-id [your VLAN] ip [your laptop IP] controller-interface vlan [your VLAN]
Create an OpenFlow instance for your vLAN
switch1(openflow)# instance "vlan[your vLAN]"

Make your vLAN a member of your OpenFlow instance
switch1(of-inst-vlan1)# member vlan [your vLAN]

Point your OpenFlow instance to your controller ID
switch1(of-inst-vlan1)# controller-id [your vLAN]

Enable OpenFlow for your OpenFlow instance
switch1(of-inst-vlan1)# enable
switch1# show openflow instance vlan2

Instance Name : vlan2
Admin. Status : Enabled
Member List : VLAN 2
Listen Port : 6634
Oper. Status : Up
Datapath ID : 00020016b96c7280
Mode : Active
Flow Location : Hardware and Software
No. of Hw Flows : 1
No. of Sw Flows : 0
Hw. Rate Limit : 0 kbps
Sw. Rate Limit : 100 pps
Conn. Interrupt Mode : Fail-Secure
Maximum Backoff Interval : 60 seconds
Probe Interval : 10 seconds

Controller Id Connection Status Connection State
------------- ----------------- ----------------
1             Connected         Active
Quick, look at smoke ping and ODL!

Why did smoke ping stop working on your vLAN?

Let’s go about fixing that using Open DayLight
Nodes Learnt

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Node ID</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>OF00:02:00:16:b9:6c:72:80</td>
<td>1 (1), 3 (3), 21 (21), 22 (22)</td>
</tr>
</tbody>
</table>

Static Route Configuration

Subnet Gateway Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Gateway IP Address/Mask</th>
<th>Node/Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>No data available</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SPAN Port Configuration
Flow Entries

There are 0 flows

Flow Name  Node

Nodes

There is 1 node

Node  Flows
None  2

Flow Details

Please select a flow
Flow Entries

There is 1 flow

<table>
<thead>
<tr>
<th>Flow Name</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow1</td>
<td>OF:00:02:00:16:b9:6c:72:80</td>
</tr>
</tbody>
</table>

Nodes

There is 1 node

<table>
<thead>
<tr>
<th>Node</th>
<th>Flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>2</td>
</tr>
</tbody>
</table>

Flow Detail

<table>
<thead>
<tr>
<th>Flow Name</th>
<th>Node</th>
<th>Priority</th>
<th>Hard Timeout</th>
<th>Idle Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow1</td>
<td>OF:00:02:00:16:b9:6c:72:80</td>
<td>500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Port</th>
<th>Ethernet Type</th>
<th>VLAN ID</th>
<th>VLAN Priority</th>
<th>Source MAC</th>
<th>Dest MAC</th>
<th>Source IP</th>
<th>Dest IP</th>
<th>TOS</th>
<th>Source Port</th>
<th>Dest Port</th>
<th>Protocol</th>
<th>Cookie</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>0x800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
fun with more than one switch (bonus section)...

select someone from your switch (you’ll be looking over their shoulder, so pick wisely)

remove vLANs from all other openflow instances, and add them to the one instance (the one belonging to the person you picked)

now there should be five switches showing up on Open DayLight. Bonus, interconnect some of the ports with a patch cable.
Review

HP switch “Virtualization” or hybrid mode.

- vLANs are invisible
- lots of interesting knobs (implementation maturing)
- Controller communications via a management vLAN

ODL “flows” tab on represents flow installed by ODL