Introduction to OpenFlow

Presented by:
Shangxin Du – TAC DC Solution

20 April, 2015
For the next 60 minutes …

Forget everything you know about:

- Transparent Bridging
- STP
- L2 Forwarding
- IP Routing
- L3 Forwarding
- And so on …
What is OpenFlow?

“standard communications interface between control and forwarding layers of an SDN architecture ...

allows direct manipulation of the forwarding plane of network devices such as physical and virtual switches and routers”

Open Networking Foundation
OpenFlow-only Switch Architecture

- Control plane is decoupled from data plane; centralized and given span of control over multiple data plane switch elements
- Controller manages switch flow table using OpenFlow protocol
  Add, update, delete flow entries, both reactively and proactively
OpenFlow-hybrid Switch Architecture

• Traditional captive control plane/data plane components co-exist with OpenFlow Switch capabilities

• OpenFlow defines logical “ports” for passing packets to traditional and OpenFlow pipelines
OpenFlow Ports

- Controller port
- Physical Port
- Logical Port (representing a VLAN)
- Logical Port (representing a VLAN)
- Logical Port (representing link bundle)
- Log. Port (Tunnel)
- LOCAL “Port”
- NORMAL “Port”

Flow Tables

OpenFlow Switch stage

Normal Switch stage
Flow Tables (OpenFlow 1.0)

- Each OpenFlow Switch has a single flow table, which is used for packet lookup and forwarding.
- Packets not matching an entry in the flow table are punted to the controller.
Flow Entries

- A flow table consists of one or more flow entries
- Each entry consists of one row in the flow table

<table>
<thead>
<tr>
<th>flow 0</th>
<th>Match Fields</th>
<th>Priority</th>
<th>Counters</th>
<th>Timeouts</th>
<th>Action(s)</th>
<th>Cookie</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flow 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flow 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flow 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flow 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- match fields: ingress port + packet headers
- priority: precedence of the flow entry
- counters: updated for each matching packet
- timeouts: maximum amount of time or idle time before flow entry expires
- action(s): one or more actions to take on match packets
- cookie: opaque data chosen by controller
Flow Table Header Fields

FLOW TABLE

<table>
<thead>
<tr>
<th>HEADER FIELDS</th>
<th>COUNTERS</th>
<th>ACTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

OF1.0 HEADER FIELDS

<table>
<thead>
<tr>
<th>Field</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress Port</td>
<td>1</td>
</tr>
<tr>
<td>Source MAC</td>
<td>2</td>
</tr>
<tr>
<td>Dest MAC</td>
<td>3</td>
</tr>
<tr>
<td>Ether Type</td>
<td>4</td>
</tr>
<tr>
<td>VLAN ID</td>
<td>5</td>
</tr>
<tr>
<td>VLAN Priority</td>
<td>6</td>
</tr>
<tr>
<td>IP SRC</td>
<td>7</td>
</tr>
<tr>
<td>IP DEST</td>
<td>8</td>
</tr>
<tr>
<td>IP Proto</td>
<td>9</td>
</tr>
<tr>
<td>IP TOS</td>
<td>10</td>
</tr>
<tr>
<td>TCP/UDP SRC</td>
<td>11</td>
</tr>
<tr>
<td>TCP/UDP DEST</td>
<td>12</td>
</tr>
</tbody>
</table>
**OpenFlow 1.0 supports a lookup into a single flow table**
Header Fields are Used for Lookup

Header fields used to build lookup key

- Lookup Key
- Data
- Data
- Data

Switch

FLOW TABLE

CPU

SWITCH FORWARDING ENGINE
Controller Writes to Switch Flow Table

If no match, Controller may receive packet and program the flow table
Forwarding Engine Forwards Packets

OPENFLOW CONTROLLER

FLOW TABLE

CPU

SWITCH FORWARDING ENGINE

Data

Data
Flow Table Counters and Actions

Several important OF1.0 Actions
Let us explore in more detail…

<table>
<thead>
<tr>
<th>FLOW TABLE</th>
<th>HEADERS</th>
<th>COUNTERS</th>
<th>ACTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADER FIELDS</td>
<td>COUNTERS</td>
<td>ACTIONS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Action #1: Forward to all ports except Input

- Switch Forwarding Engine
- CPU
- Flow Table
- OpenFlow Controller

Packet flow diagram:
1. Packet enters the Switch Forwarding Engine
2. Match in Flow Table
3. Forward to all ports except Input

Diagram shows the interaction between the Switch, CPU, Flow Table, and OpenFlow Controller.
Action #2: Redirect to Controller
Action #3: Output packet from Controller
Action #4: Forward to Local CPU
Action #5: Rewrite Packet Headers

1. Openflow Controller
2. Flow Table
3. Switch Forwarding Engine
4. CPU
5. Packet

Switch
Action #6: Forward to Input Port
Action #7: Forward to Destination Port
Action #8: Drop Packet
Important OF1.0 Actions: Summary

**Important Actions**

1. Forward to all ports except input
2. Redirect to Controller
3. Output from Controller
4. Forward to local CPU
5. Rewrite Packet Header
6. Forward to input port
7. Forward to dest port
8. Drop packet
OpenFlow 1.1

- Single flow Table
- L2, IPv4 focused matching
- Basic actions

- Multiple flow tables
- Group table
- Packet processing options
- MPLS
- VLAN
OpenFlow 1.2

- Single flow Table
- L2, IPv4 focused matching
- Basic actions

- Group table
- Multiple flow tables
- Packet Processing Options
- MPLS, VLAN

- IPv6
- Extensible matching
OpenFlow 1.3.x

- Single flow Table
- L2, IPv4 focused matching
- Basic actions

- Group table
- Multiple flow tables
- Packet Processing Options
- MPLS, VLAN

- IPv6
- Extensible matching

- IPv6 Extended Headers
- Meter table
- Auxiliary Connections
- Advanced MPLS
- PBB
- Version negotiation
- Controller connections

---|---|---|---|---|---
OF 1.0 | OF 1.1 | OF 1.2 | OF 1.3.0 | OF 1.3.1 (stability) | OF 1.3.2

© 2013 Cisco and/or its affiliates. All rights reserved.
OpenFlow stability, OpenFlow 1.4 and up

OF 1.0 stability Cisco
OF 1.1
OF 1.2 stability Cisco
OF 1.3
OF 1.4
OF 1.5

Aug 2013
in progress

• Table sync
• Non-Ethernet
• Flow monitoring

• Generic tunnels
• “Split brain”
OpenFlow @ Cisco
OnePK Architecture

- C, JAVA Program
- onePK API Presentation
- onePK API Infrastructure
- IOS / XE (Catalyst, ISR, ASR1K)
- NX-OS (Nexus Platforms)
- IOS-XR (ASR 9K, CRS)
OpenFlow Agent with onePK Architecture

- Application -> OpenFlow Agent
- onePK API Presentation -> Native NOS and Container
- onePK API Infrastructure -> Native NOS
- IOS / XE (Catalyst)
- NX-OS (Nexus Platforms)
- IOS XR (ASR 9K)
OpenFlow Logical Switch(es)
Hybrid OpenFlow on Cisco devices

Pure OpenFlow*
- All ports are OpenFlow only
- All forwarding decisions by Controller
- Example: Network Monitoring

* There still may be some non OpenFlow ports for connecting to management device or initial setup

Ships in the Night
- OF co-exists with normal forwarding
- Port segregation
- No traffic flow between domains
- Example: Network Slicing

Integrated
- Packet may traverse OpenFlow pipeline and normal pipeline
- Output to NORMAL and Logical ports
- Example: Traffic Steering
Use Cases
Use Case: Data Center Monitoring

Visibility Tools

- Java and RESTful
- Wireshark
- Video Monitor
  - Dynamic Filter and Forwarding Event Driven / Real Time

Production Network

- With XNC Monitor Manager Solution
- OpenFlow Enabled Nexus 3000s
- SPAN Ports
- OPTICAL TAPS

Replaces Matrix Network with Nexus 3000s, Controller, and Monitor Manager App
Objective:
Establish a flow to bypass the firewall for trusted traffic to improve application performance.
Data Center Optimization
Transit Selection: Network Parameter Driven (Latency)

Other Transit Selection Parameters Also Possible
Control egress path based on application requirement and priority
Simple forwarding rules help improve WAN utilization
Service Provider
Traffic Engineering with PCE and OpenFlow

1. Path Request from Application
2. Tunnel Create Request
3. RSVP Signaling
4. State Report
5. Traffic steering

Existing Functionality
New Functionality
OpenDaylight

- A Linux Foundation Project
- Industry-Supported
OpenDayLight
OpenFlow Packet

- Frame 316: 1186 bytes on wire (9488 bits), 1186 bytes captured (9488 bits)
- Internet Protocol Version 4, Src: 10.75.45.138 (10.75.45.138), Dst: 10.75.45.137 (10.75.45.137)

**OpenFlow 1.3**

- Version: 1.3 (0x64)
- Type: OFPT_MULTIPART_REPLY (19)
- Length: 1120
- Transaction ID: 1684
- Type: OFPMP_FLOW (1)

**Match**

- Type: OFPMT_OXM (1)
- Length: 24
  - OXM field
    - Class: OFPPM_OPENFLOW_BASIC (0x8000)
    - 0000 100 = Field: OFPMP_DFO_ETH_SRC (4)
      - 00:00 = Has mask: false
    - OXM field
      - Class: OFPPM_OPENFLOW_BASIC (0x8000)
      - 0000 011 = Field: OFPMP_DFO_ETH_DST (3)
        - 00:00 = Has mask: false
  - Instruction
    - Type: OFPIT_APPLY_ACTIONS (4)
      - Length: 24
      - Pad: 00000000
  - Action
    - Type: OFPAT_OUTPUT (0)
      - Length: 16
      - Port: 2
      - Max length: OFPCHL_NO_BUFFER (0xffff)
      - Pad: 00000000000000
Thank you.