OpenFlow network virtualization with FlowVisor
Research Project 2

Sebastian Dabkiewicz
System and Network Engineering
University of Amsterdam

17th October 2012
1. OpenFlow

2. FlowVisor

3. Research Question

4. Experiments & Results

5. Conclusion

6. Questions
OpenFlow is a form of Software Defined Networking (SDN)

Control plane moved to an external controller

Traffic between switch and controller = OpenFlow protocol
Switch vs. OpenFlow switch
Flows

**Match**
- Ingress port
- Ethernet source/destination address
- Ethernet type
- VLAN ID
- VLAN priority
- IPv4 source/destination address
- IPv4 protocol number
- IPv4 type of service
- TCP/UDP source/destination port
- ICMP type/code

**Action**
- Forward
- Enqueue
- Drop
- Modify field
Network Virtualisation with FlowVisor

- Developed at Stanford
- Slices network
- Multiple virtual networks on a switch
- Multiple OpenFlow controllers per switch
- A bit like VLANs but more advanced
- Transparent proxy between the OpenFlow switch and OpenFlow controller
- Configuring done with the `fvctl`-tool
Slice

- Slicename
- Controller URL
- E-mail

Example

```
fvtcl createSlice slice1 tcp:145.100.37.143:6633 slice-1@example.com
```

FlowVisor command  Slicename  Controller URL  Admin e-mail
FlowSpace

Slice policy rule

Classifies incoming traffic

Based on:

- DPID
- PRIORITY
- FLOW_MATCH
- SLICE_ACTIONS
FlowSpace DPID & PRIORITY

DPID DataPath IDentifier
- 8 hex octets
- example 00:00:00:23:10:35:ce:a5

Priority
- Range 0-2^{31}
- Highest match
FLOW_MATCH:
Like normal OpenFlow match

SLICEACTION:
- $\text{DELEGATE}=1$, ability to delegate control to other slice.
- $\text{READ}=2$, read messages that belong to a slice.
- $\text{WRITE}=4$, same as read, but also able to change flows.
Example

fvctl addFlowSpace 00:00:e8:9a:8f:fb:c3:5b 100 in_port=3 Slice:floodlight1=4

FlowVisor command  |  DPID  |  Priority  |  Match  |  Sliceaction
Research Question

Is the current FlowVisor implementation (ver. 0.8.5) suitable to create stable virtual networks in production environments?
Port Based Slices

Slice 1: port 1, port 2
Slice 2: port 3, port 4

Experiment:
Let the controller from Slice 2 push a Flow for Slice 1.
<table>
<thead>
<tr>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:15:53.208 [New I/O server worker #1-1] ERROR n.f.core.internal.Controller - Error OFPET_FLOW_MOD_FAILED OFPFMFC_EPERM from OFSwitchImpl [/145.100.37.143:55771 DPID[00:00:e8:9a:8f:fb:c3:5b]]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error</th>
</tr>
</thead>
</table>
| OFPET_FLOW_MOD_FAILED  
OFPFMFC_EPERM |
VLAN Slices

Slice 1: port 1, port 2, VLAN 50
Slice 2: port 3, port 2, VLAN 60

Experiment:
Share a port based on VLANs.
Worked fine :-}
Switch events

Slice 1: port 1, port 2
Slice 2: port 3, port 4

Experiment:
Trigger port change by unplugging the network cable on port 2.
Switch events - Result 1
Switch events - Result 2

Result - FlowVisor

1 Oct - 13:33:38 INFO
org.flowvisor.log.AnyLogger.log(AnyLogger.java:38) modifying port 2

Result - OpenFlow Controller

13:33:38.057 [New I/O server worker #1-1] DEBUG n.f.core.internal.Controller - Port #2 modified for OFSwitchImpl [/145.100.37.143:42990 DPID[00:00:e8:9a:8f:fb:c3:5b]]
13:33:38.059 [pool-3-thread-13] DEBUG n.f.d.internal.DeviceManagerImpl - Attachment point changed for device: Device [deviceKey=1, entityClass=DefaultEntityClass, MAC=00:1c:73:08:11:8d, IPs=[], APs=[]]
Switch events - Result 3

Frame 388: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits)
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 145.100.37.143 (145.100.37.143), Dst: 145.100.37.143 (145.100.37.
Transmission Control Protocol, Src Port: 57364 (57364), Dst Port: 6635 (6635), Seq: 3891, Ack: 1369
OpenFlow Protocol
- Header
  - Port Status
    - Reason: Some attribute of the port has changed (2)
    - Physical Port
      - Port #: 10
      - MAC Address: QuantaCo_fb:c3:5b (e8:9a:8f:fb:c3:5b)
      - Port Name: ge-1/1/10
    - Port Config Flags
  - Port State Flags
    - No physical link present: 1
    - STP state: Not learning or relaying frames
      - Port Current Flags
      - Port Advertised Flags
      - Port Supported Flags
      - Port Peer Flags

0050 00 00 00 0a e8 9a 8f fb c3 5b 67 65 2d 31 2f 31 10 00 00
0060 2f 31 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 01 00 00 04 00 00 00 04 2f 00 00 0e 2f 00 00
0080 04 2f
Slice Based on destination MAC-address

Slice 1:
dst-mac: aa:aa:aa:aa:aa:aa,

Slice 2:
dst-mac: cc:cc:cc:cc:cc:cc

Experiment:
Slice Based on destination MAC-address. Put a IP-based Rule to the Switch.
mac rules worked

IP rules also which shouldn’t be the case.

<table>
<thead>
<tr>
<th>Result - FlowTable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowTable:</td>
</tr>
<tr>
<td>ip,nw_dst=192.168.1.1 actions=output:1</td>
</tr>
</tbody>
</table>
Conclusion

- I would not recommend to use FlowVisor in a production environment at the moment.
- Bugs? (still) present.
- Missing Documentation.
- Simple usage i.e. Port based OK.
- Complex setups should be well tested.
Questions?