OpenFlow Virtualization Framework with Advanced Capabilities

Balázs Sonkoly, András Gulyás, Felicián Németh, János Czentye, Krisztián Kurucz, Barnabás Novák, Gábor Vaszkun

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Email: {sonkoly,gulyas,nemethf}@tmit.bme.hu
vaszkun@gmail.com
Background

- Network virtualization in OpenFlow by FlowVisor
  - proxy-based virtualization
  - simple slicing
  - policy control to switch resources
  - stolen figs…
Motivation & Goals

• FlowVisor (proxy) based network virtualization
  – limits switch model & capabilities
  – depends on OpenFlow protocol & switches (currently v1.0)
  – FlowVisor should be modified if
    • modified matching
    • new OF messages
    • novel forwarding functions

• OpenFlow
  – OF-CONFIG out, but still has version questions
  – current tools: only running/configuring data plane & FlowVisor

• Goals: novel virtualization framework
  – add heavy-weight virtualization to OpenFlow
    • enhancing rapid prototyping & testing
    • maintain the physical network
    • provision virtual slices
  – support
    • multiple versions of OF protocols & switches (v1.0, v1.1)
    • switches with novel forwarding capabilities
    • different controllers
  – design OpenFlow management framework
    • switch management
    • controller management
Our framework
Development Environment

- Open-source NETCONF implementation
- Easy to use data modeling language – Yang

**netconfd** – server app
**yangcli** – client app
**yangdump** – code generator
Data model

Configuration example:
of-instance 2 {
  id 2
  config {
    desc '1.0 of switch 1'
    status running
    dp-pid 1505
    of-pid 1552
    ofdatapath-path /usr/bin/of10/ofdatapath
    ofprotocol-path /usr/bin/of10/ofprotocol
    dp-of-socket /tmp/dp2.sock
    eth-if eth1
    eth-if eth2
    controller-address 192.168.213.230
    controller-port 6635
    controller-type tcp
  }
}

QoS extension

- RPC-s to handle basic QoS queue creation and removal

- OF switches up to version 1.1 only supports min-rate

- Controller app capable to use the defined queues is under development
Integration with NMS

- Open-source network management system
- Easy to use surface
- Integration with Yuma’s client program, yangcli
Topology discovery module

- Integrated with OpenNMS
- OpenNMS and OF controller communicates through HTTP
- Topology viewer uses Google Web Toolkit
- Discovery module based on the NOX controllers discovery app
OpenWrt implementation

- Framework is compilable on SOHO routers
  - Device needs an open operating system, such as OpenWrt
  - Cross-compiling needed

- Software based OF Ref. Switch’s performance still an issue
  - Running multiple switch instances can degrade a Gbits/s interface’s performance to 50 Mbits/s or even further
physical links
virtual links
(veth)
management
OF control
interfaces
management interfaces
QoS demo traffic

```
rpc-reply {
    queues 3 1 {
        port 3
        q_id 1
        bandwidth 6
    }
    queues 3 2 {
        port 3
        q_id 2
        bandwidth 4
    }
}
```
Conclusions & Future works

• We defined a virtualization framework which could replace the FlowVisor structure
• Extension of the framework with QoS
• Visualization of each virtual/physical topology

Ongoing/Future works:

– NOX controller applications:
  • Dynamic use of QoS queues
  • Dynamic mapping switch configuration
– OF-CONFIG 1.0 and 1.1 implementation for software based reference switches with the help of Yuma-tools