A Datapath-centric Virtualization Mechanism for OpenFlow Networks

R. Doriguzzi Corin\textsuperscript{1}, E. Salvadori\textsuperscript{1}, M. Gerola\textsuperscript{1}, M. Suñ\textsuperscript{e}e\textsuperscript{2}, H. Woesner\textsuperscript{2}

\textsuperscript{1}CREATE-NET(I), \textsuperscript{2}BISDN(D)

EWSDN Workshop, Budapest, Hungary

September 1, 2014
• Motivations: current status of the OpenFlow testbeds
• Software architecture of the proposed framework
• Implementation
• Evaluation
• Conclusions
• Next steps
Typical configuration:
- OpenFlow switches
- Virtualization servers (computational resources)
- FlowVisor to slice the switches
- Experiment orchestration software

GENI
(USA - www.geni.net)

OFELIA
(EU/Brazil - www.fp7-ofelia.eu)

GÉANT
(EU - www.geant.net)
Network virtualization in OF

The most popular OpenFlow tool for NV is FlowVisor

• FlowVisor exposes different views (slices) of the physical infrastructure to different controllers
• FlowVisor is the reference tool for assigning virtual networks to experimenters in GENI, OFELIA and GÉANT

Many other NV mechanisms leveraging on OpenFlow are being investigated
FlowVisor in a nutshell

1. FlowVisor needs to inspect the OpenFlow protocol to ensure that each Controller receives only the messages that match its slice policy.
2. FlowVisor rewrites the messages to enforce the isolation between slices.

- **Expedient** connects to different aggregate managers and provides a GUI.

- **Opt-In/FOAM**: aggregate managers for OF resources.

- **FlowVisor** slices the flowspace.
FlowVisor’s limitations

- **FlowVisor only supports version 1.0 of the OF protocol**
  - No way for the research community to explore and test the newest functionalities of the protocol in realistic environments such as GENI, OFELIA and GÉANT

- **FlowVisor represents a SPoF for the testbeds**
  - A failure of one instance of FV will bring down all the experiments running on the aggregate

- **Latency overhead**
  - FV acts as a proxy on the control channel. Therefore it establishes additional TLS connections to forward the OF messages from the switch to the controllers and vice-versa
A distributed virtualization approach

The Virtualization Agent (VA) runs on the switches and performs the flowspace slicing operations.

The Virtualization Agent Orchestrator (VAO) is a JSON-RPC client used to configure the VA instances.
Virtualization Agent
Aims to address the aforementioned FlowVisor’s limitations:
• No Single Points of Failure (no proxy on the control channel)
• Support for multiple versions of the OpenFlow protocol
• Reduced latency overhead on the control channel

Virtualization Agent Orchestrator
Its northbound interface is compatible (super-set of functions) with existing control software such as FOAM (the FlowVisor OpenFlow Aggregate Manager). This allows an easy integration with existing Control Frameworks.
The Virtualization Agent is implemented as a plug-in for xDPd (www.xdpd.org)

What is xDPd?
“The eXtensible Datapath daemon (xDPd) is a multi-platform, multi OpenFlow version, open-source data path, built focusing on performance and extensibility.”

Why xDPd?
• Is available for several hardware platforms (x86, OCTEON, NetFPGA 10G, Intel DPDK, EZAppliance, Broadcom)
• Provides stable support for OpenFlow versions 1.0, 1.2 and 1.3
• Includes a plug-in management sub-system to extend the control plane functionalities
VA implementation (2)

Includes: HW-specific layers (forwarding module, drivers) and an abstraction layer.

Platform-independent layer. Includes the plug-in subsystem, the Logical Switches management and the OpenFlow endpoints.
**Goal:** to compare the VA against FlowVisor in terms of latency introduced on the control channel by the slicing operations.
Results

EWSDN 2014 – R. Doriguzzi Corin

![Graph showing cumulative probability against new flow latency (ms) for different scenarios: xDPd, xDPd + VA, xDPd + FV.](image)

- xDPd avg. latency: 0.477ms
- xDPd + VA avg. latency: 0.577ms
- xDPd + FV avg. latency: 1.006ms
Conclusions

• The proposed framework is based on a distributed virtualization architecture that is able to run on multi-version OpenFlow switch scenario.

• It also provides management interfaces that are compatible with existing experiment orchestrations tools.

• The final goal of the framework is to allow devices running OpenFlow v1.1+ to be shared and tested in realistic environments.

• Preliminary results demonstrate that, comparing with the current deployments, the proposed mechanism can be adopted without degrading the performance of the control channel.
Next steps

Multiple tables management (the GOTO_TABLE instruction)
  • Partitioning the flow tables of the switches (scalability?)
  • Packet tagging (transparency?)

Management of the group entries: reserve some bits of the group identifier to assign the group to the Virtual Network
Questions?

Email: roberto.doriguzzi@create-net.org