Carving Research Slices Out of Your Production Networks with OpenFlow

Rob Sherwood
† DT Inc. R&D Lab
5050 El Camino Real
Los Altos, CA USA
robert.sherwood@telekom.de

Glen Gibb
⋆ Stanford University
351 Gates Building
Palo Alto, CA USA
grg@stanford.edu

Masayoshi Kobayashi
△ NEC Labs
Tokyo, Japan
mkobaya1@stanford.edu

Categories and Subject Descriptors
C.2.2 [Computer Systems Organization]: Computer-Communication Networks—Network Architecture and Design

General Terms
Management, Design, Experimentation

Keywords
Flowvisor, OpenFlow, Architecture, Virtual networks, Slicing

1. SLICED PROGRAMMABLE NETWORKS

OpenFlow [6] has been demonstrated as a way for researchers to run networking experiments in their production network. Last year, we demonstrated how an OpenFlow controller running on NOX [4] could move VMs seamlessly around an OpenFlow network [1]. While OpenFlow has potential [3] to open control of the network, only one researcher can innovate on the network at a time. What is required is a way to divide, or slice, network resources so that researchers and network administrators can use them in parallel. Network slicing implies that actions in one slice do not negatively affect other slices, even if they share the same underlying physical hardware. A common network slicing technique is VLANs. With VLANs, the administrator partitions the network by switch port and all traffic is mapped to a VLAN by input port or explicit tag. This coarse-grained type of network slicing complicates more interesting experiments such as IP mobility or wireless handover.

Here, we demonstrate FlowVisor, a special purpose OpenFlow controller that allows multiple researchers to run experiments safely and independently on the same production OpenFlow network. To motivate FlowVisor’s flexibility, we demonstrate five network slices running in parallel: one slice for the production network and four slices running experimental code. Our demonstration runs on real network hardware deployed on our production network at Stanford and a wide-area test-bed with a mix of wired and wireless technologies.

2. FLOWVISOR ARCHITECTURE

Architecturally, FlowVisor acts as a transparent virtualization layer between OpenFlow switches and controllers. Network devices generate OpenFlow protocol messages, which go to the FlowVisor and are then routed by network slice to the appropriate researcher(s) (Figure 1). OpenFlow messages from researcher controllers are vetted by the FlowVisor to ensure that the isolation between slices is maintained before being forwarded to switches. Thus, the FlowVisor appears as a virtual controller to the switches and as a network of virtual switches to the researcher controllers. FlowVisor is intentionally architecturally neutral: it makes no assumption about the function or operation of the switches or controllers, save that they speak OpenFlow. We architect FlowVisor as a transparent layer for three reasons:

Centralized policy enforcement. All control traffic, from switch to controller and from controller to switch, traverses the FlowVisor. This provides FlowVisor a complete view of the network’s state and allows it to enforce policy by dropping or rewriting OpenFlow control messages. Additionally, centralizing policy decisions makes it easier to reason about the set of allowable actions and debug errors should they occur.

Recursive delegation is the ability to re-delegate control of a subset of a network slice. Because FlowVisor acts as a transparent proxy, it is possible to cascade FlowVisor instances, making recursive delegation trivial. We expect recursive delegation will be an important property for virtual networks as it eases network administration overhead and improves resource allocation.

Decouple control and virtualization technologies. Rather than building virtualization support directly into the OpenFlow protocol itself, we intentionally keep the control and virtualization aspects orthogonal. This allows each technology to evolve independently, avoiding new forms of ossification.

FlowVisor defines slices along any combination of ten packet header fields, including physical layer (switch ports), link layer...
OpenPipe’s traffic stressed the FlowVisor flexible FlowSpace slicing in terms of its ability to slice by ethernet type.

4. ADDITIONAL AUTHORS

A small army of authors contributed to this demo (refer to title for affiliations):

Michael Chan  Adam Covington† Mario Flajslik
Nikhil Handigol† Te-Yuan Huang † Peyman Kazemian†
Jad Naous † Srinivasan Seetharaman† David Underhill*
Tatsuya Yabe † Kok-Kiong Yap † Yiannis Yiakoumis†
Hongyi Zeng† Guido Appenzeller † Ramesh Johari†
Nick McKeown† Guru Parulkar†

5. REFERENCES