Whitepaper | OpenMUL

PRISM Application Overview

Authors: *Dipjyoti Saikia,*
Nikhil Malik

September 2014

Contents

Section 1

The SDN promise

One of the software defined networks substantial benefits is to provide "centralized management and control of networking devices from multiple vendors". Since its inception Openflow has sought to increase network functionality while lowering the costs associated with operating networks. While many have tried to completely reinvent the networks with Openflow and related technologies, the community has had less success to take up the reins of what is available in the legacy networking world today and improving them with new tools like Openflow.

This whitepaper will discuss use case(s) of Openflow based IP network which leverages existing protocols like OSPF, BGP and tries to incrementally improve upon them by providing a single management plane. It will also explore how white-boxes can be employed with resilient controller software to improve upon existing networking devices/routers which sit at home as a small home gateway router to big core network routers using plain vanilla Openflow. The growing concerns of scalability and performance with SDN will also be addressed in detail.

Section 2

The architecture of Internet Routers

Let us recap by having a quick look at common functions of Internet routers which are ubiquitous in current networking landscape.

Most of the routers are usually implemented as a separate control and data (or forwarding) planes. Depending on capacity and function IP routers vary in shape and sizes. Many of them are implemented as a single box while others can span multi-chassis with multiple line cards.

The major job of IP routers is to perform IP forwarding and routing. There is a lot of additional functionality built into these devices but as deployment moves from edge of the network to the core, the functionality takes a back seat and performance becomes the key factor.

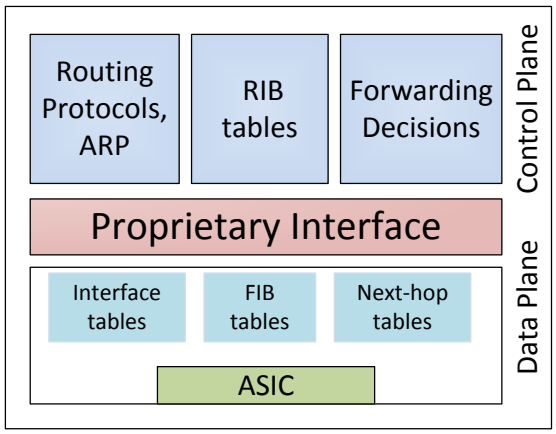

Figure 1. Basic blueprint of an IP router

From the data plane perspective, the most basic IPv4 forwarding functions performed by a router are categorized below:

Forwarding Function	Supported by Openflow
Interface MAC check	Yes
Routing Lookup	Yes
Reverse Path Check	Yes
TTL decrement	Yes
Next hop MAC address rewrite	Yes
Source MAC address rewrite	Yes
Output packet to a port	Yes

Table 1. Basic Forwarding functions

From the control plane perspective, an IP router supports many standard routing protocols mainly OSPF, BGP, RIP, LDP, RSVP etc. These protocols have stood the test of time and available as open-source projects like Quagga, XORP as well as commercially from different vendors.

Section 3

The major problem

The major problem with legacy network equipment which spawned the SDN movement has been the lack of simple and centralized management plane all the while making the network highly agile. Various legacy devices like IP routers had different and proprietary control plane to data plane interfaces and hence it became impossible to centralize the control plane. Even the north bound interfaces on top of control planes like SNMP, netconf varied wildly for each device.

One of the major USP of Openflow has been standardized interface between control and data plane. It gives us immense potential to centralize control planes and provide centralized control to network operators.

Having such a separation is great but is it a good idea to reinvent the way networks all around the world have been designed?

The simple answer-we don't need to redesign the networks. Because networks still need to talk, say, OSPFv2 or BGPv4. These are built into DNA of modern network infrastructure. But, Openflow can be definitely be used to solve the centralized management problems. Effort should be put to bring more devices under centralized management umbrella while letting interface to the external world unchanged i.e. use OSPF, BGP etc as is.

Section 4

The PRISM approach

PRISM (Perfect Routing Integration of SDN using MuL) takes a modular approach to solve the problem outlined in **Section 3**. In a nutshell, it lets a single control plane routing instance run as is across a set of Openflow devices bringing them under a centralized management plane. With the advent of white-box networking, it provides a compelling use-case of creating a solid routing and forwarding entity by employing devices and software having completely open interfaces.

The various components of PRISM are shown in **Figure 2.**

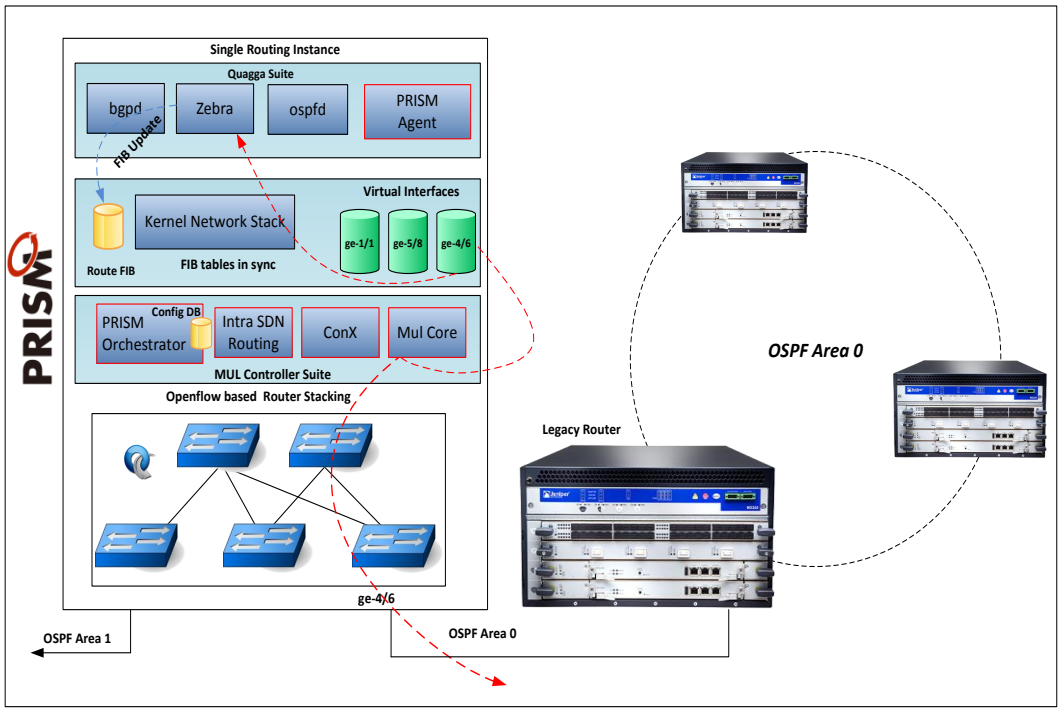

Figure 2. PRISM architecture

The centralized routing instance runs separately from the cluster of data forwarding switches in a physical server or as a virtual machine. Routing instance can be open-source routing stacks like Quagga, XORP or any commercially available software with proper integration. Once routing instance learns the external routes, PRISM implements application aware routing and stiches the traffic-engineered data path by using various services of MUL SDN controller platform. MUL platform completely segregates flows derived using external routes from internal fabric switching flows thereby achieving high

scalability and preserving precious flow table sizes in internal Openflow switches. Overall, PRISM lets network traffic to be more controlled, leads to better utilization of resources and better network operations, makes management and scalability of the network to be easier than it ever was. The implementation preserves external interfaces (OSPF, BGP etc) as used by various network devices today while delivering cutting-edge innovation at the same time.

MUL Controller's 'ConX' service takes care of managing the internal fabric path. It can maintain various active and backup paths for the internal path thereby achieving highly reliable and lossless fast path recovery. It uses advanced Openflow 1.3 features such as fast-failover groups to maintain the internal switching fabric and helps mimic traditional backplanes of legacy IP routers. PRISM architecture does full justice to feature set provided by Openflow1.3 and higher.

PRISM Orchestrator gets the resolved next hop information of the legacy network attached to edge nodes and instructs ConX to stitch path from a edge node to all other edge nodes of the Openflow Island in one singe transaction. ConX uses another service "Topo-Routing" which exposes path-finding service to its clients.

Topo-Routing, another fine arrow in the quiver, gathers the neighborhood information. As a part of centralized system, this service can see the complete topology of Openflow fabric. This service uses LLDP (Link layer Discovery Protocol) to find the topology of Openflow Island. It uses Floyd-Warshall algorithm to find the unique optimized paths from one node to another. It supports N-way ECMP as well as loop detection. As the network has become more agile and dynamic, its management has become more cumbersome. An application which is highly scalable, helps configuring the data path and maneuvering the data through optimized data path is the need for the hour.
PRISM shapes the idea of having a separate centralized control plane from data path elements for application-aware routing and put the vision to reality. Resiliency, hitless forwarding, modular approach, integrity, flexibility and ease of maintenance are its salient features.

Section 5

The challenges

Scalability of PRISM solution presents unique challenges because it has to support large number of devices and centralize the control plane to provide single management plane benefits. On top of that, Openflow switches traditionally support limited full-match flows. PRISM uses many innovative approaches to achieve desired scalability numbers.

Flow table optimization

Many Openflow controller based solutions use the complete Openflow Island as a single domain and path-stitching happens by installing flows in all switches in a path. This has far flung and adverse consequences like the need to re-program all flows in a path whenever path needs recalculation during events like active link going down. PRISM and MUL controller segregate the flows such that only the flows corresponding to the external routes need to be programmed on the edge switches. The internal switches need not be aware of any these flows. MUL controller provides a dedicated flow connector module 'ConX' which maintains this segregation and hides all complexity from the different applications.

Route update handling

Route-convergence time is another important aspect of routing software design and PRISM needs to optimize route update handling. Its architecture leverages various Openflow features to optimize convergence times. It table-types various tables offered by Openflow devices and models them according to routing requirements.

Multiple routes can be associated with single Next Hop so keeping the flows for route and associated next hop in a single table will add complexity when it comes to any change in next hop. All the flows need to be updated and update time will depend upon the number of associated routes. But Orchestrator installs the flows for routes and its associated next hop in two separate tables. Whenever there is change in the next hop entry, only a single flow needs to be updated in the table dedicated for next hop. Hence, usage of multiple tables reduces unnecessary flow updates and makes the maintenance easier.

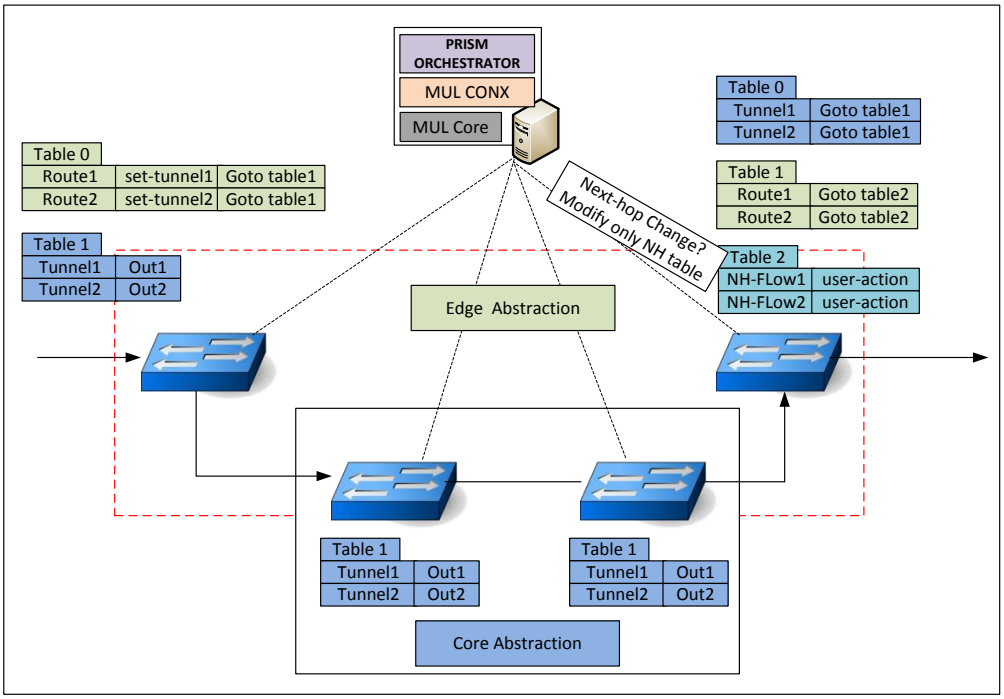

Figure 3. PRISM flow-table optimization

Non-Blocking backplanes

Traditional router design puts a lot of emphasis on router back planes for multi-line card routers. Many modern routers use crossbar switches to achieve non-blocking high performance forwarding. Since PRISM tries to leverage off-the shelf components, it doesn't have access to highly specialized switching backplanes. To provide forwarding performance needed at scale, PRISM uses an internal switching fabric modeled using modern data-center scale out leaf-spine design to achieve overall performance similar to high-performance routers available today.

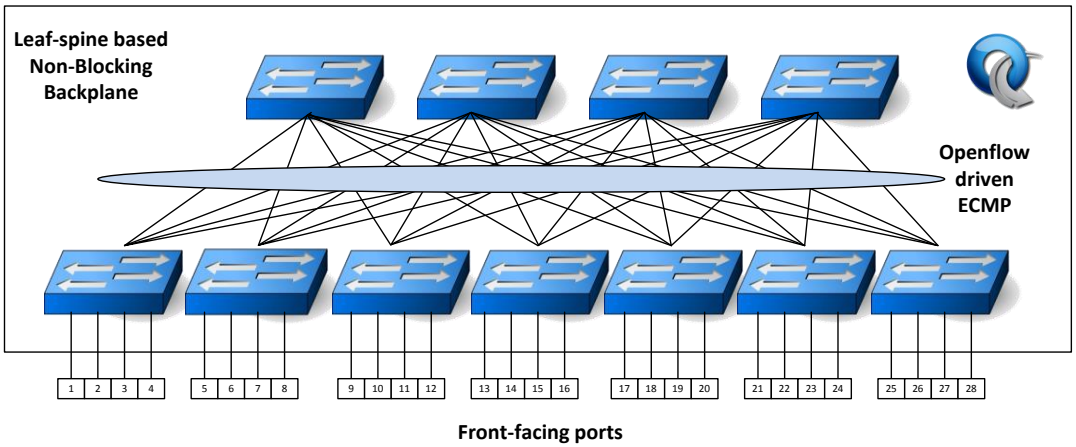

Figure 4. PRISM non-blocking forwarding

The ARP problem

Due to centralized design of Openflow controllers, PRISM needs to minimize the effect of ARP processing in the control plane. Forwarding plane has to punt packets to controller for ARP processing when next-hop mac-addresses are not resolved. Controller software simply can't handle traffic at line rate. PRISM tries to alleviate this using rate limiting non-control plane traffic send to the controller in the switches. However, control plane ARP handling usually done at OS kernel (e.g. Linux kernel) ages the ARP entries. If Linux detects that ARP entries are somehow unused, which it does in our case since after initial installation, forwarding happens at the switches, it simply deletes it. Tearing and re-provisioning the data path can be very costly if data rate is high.

PRISM orchestrator employs state-of-art design by partially simulating the state machine of Linux and ensures that active next hop entries in Kernel never remain STALE thus offloading Linux from checking next hop liveliness. Orchestrator checks if a particular next hop is alive using Openflow statistics and syncs information with Linux making sure hot ARP entries are never deleted from the system.

Control plane failures

Resiliency and Integrity are two of the important factors which make PRISM stand right in front of the league. PRISM architecture takes care about the unwanted scenarios like link failures, PRISM application/services restart and control plane restart.

ConX and Topo-Routing modules takes care of link failures or link modifications by detecting them and dynamically making necessary changes in the internal data path, hence keeping the integrity of the Openflow Island intact. In case of link modification, a loop may be introduced in the network which can become fatal for the network itself but Topo-Routing module has the capability to detect the loops and immediately takes the action to avoid such situation.

In case, any of PRISM application and services restart then PRISM's design ensures that data forwarding in the Openflow fabric remains intact and always in synchronized state with the system and network. PRISM also supports the peer routing engine's graceful restart features to provide non-stop forwarding.

Section 6

Use-Cases

The white-box SDN router

An IP router is a must have entity in any network. Any white-box switch combined with PRISM helps get a plug-and-play feature-rich IP router up and running in very less time. If there is a need to scale-up, more white-boxes can be attached to the basic set. One can also take down a white-box to scale-down and apply it to any other purpose without the need to take down the whole network. Considering the rising cost of legacy router interfaces, PRISM provides a cost-effective way to build an IP network out of off-the-shelf components supporting open interfaces.

Internet exchange points

An Internet exchange point (IXP) enables local networks to efficiently exchange information at a common point within a country rather than needing to exchange local Internet traffic overseas. Therefore, an IXP is a component of Internet infrastructure that can increase the affordability and quality of the Internet for local communities. IXP's have been traditionally implemented with traditional L2 (and at times L3) equipment. PRISM can be readily used as a replacement in legacy IXP's and bring cost-effectiveness in terms of CAPEX and OPEX which are very important factors in running an IXP. PRISM allows easy integration of value added services like multicasting, private-interconnects, CDN, route-collectors etc.

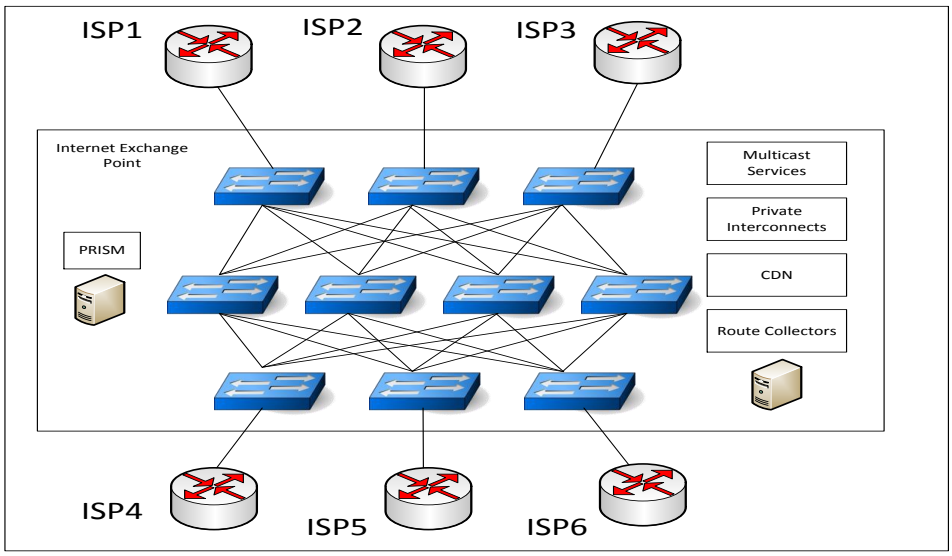

Figure 5. PRISM in IXPs

Date-Center L3 demarcation

Many Greenfield data-centers tend to use all-out Openflow based fabric solutions to gain benefits of software defined networking. Many Openflow fabric solutions exist today for seamless east-west traffic flow but when it comes to north-south traffic, there is no elegant Openflow based L3 demarcation point where SDN/Openflow meets the external routed world. Either we have to rely on inflexible hybrid Openflow-legacy switches, use external software or hardware routers. PRISM gives the option to deploy solid pure Openflow based L3 gateways to the software defined data-center.

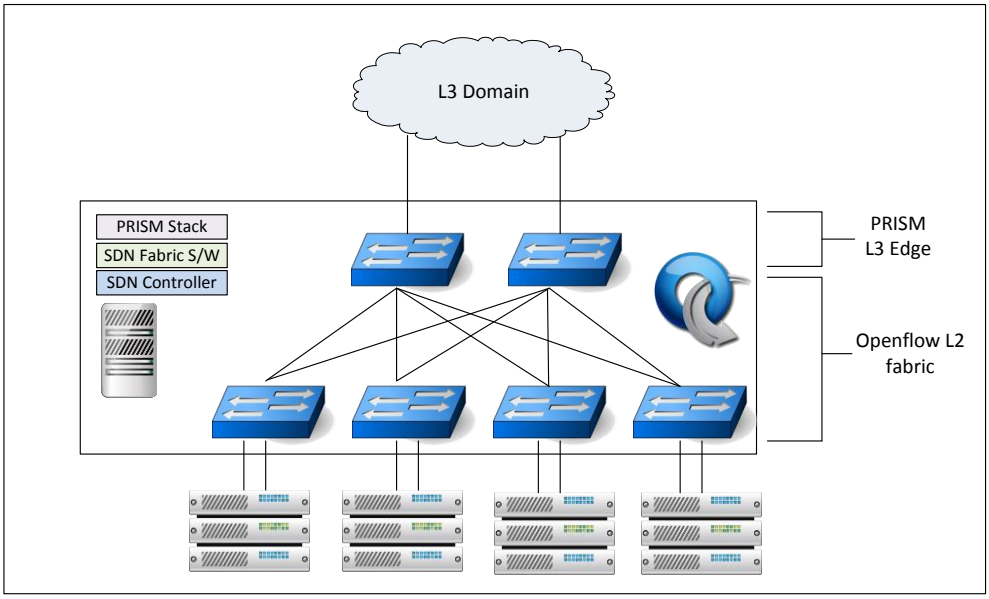

Figure 6. PRISM as L3 edge in Data-Centers

Section 7

Conclusion

In this paper we introduced a powerful new software approach to abstract legacy IP routing and forwarding using Openflow. We covered various aspects of such design and use-cases of the technology in various areas. PRISM is available today as beta release and has been integrated with Openvswitch-2.3.0. Future work would be to integrate with more vendor devices as well with Openstack APIs to provide data-center gateway functionality.

Section 10

Bibliography

1) Openflow Specifications
   https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.3.pdf

2) Openflow integration approach from Juniper
   - http://static.techfieldday.com/wp-content/uploads/2011/10/jnpr-dward.pdf

3) RouteFlow: Control planes running as separate VMs using Openflow
   - https://sites.google.com/site/routeflow/home

4) Brocade hybrid Openflow port mode
   - http://www.brocade.com/solutions-technology/technology/software-defined-networking/openflow.page

5) Openflow deployment models
   - http://blog.ipspace.net/2011/11/openflow-deployment-models.html

6) SDX: A Software Defined Internet Exchange
   - http://www.cs.umd.edu/~dml/papers/sdx-ons13.pdf