Seamless Interworking of SDN and IP

Pingping Lin[†]   Jonathan Hart[‡]   Umesh Krishnaswamy[‡]   Tetsuya Murakami[#]   Masayoshi Kobayashi[*]   Ali Al-Shabibi[‡]   Kuang-Ching Wang[§]   Jun Bi[†]

[†] Institute for Network Sciences and Cyberspace, Tsinghua University
[†] Tsinghua National Laboratory for Information Science and Technology
[‡] Open Networking Laboratory, Menlo Park, CA, USA
[*] NEC Corporation of America;   [#] IP Infusion;   [§] Clemson University

Categories and Subject Descriptors

C.2 [**Computer-communication networks**]: Network Architecture and Design.

Keywords

Software Defined Networking, SDN-IP Network Peering.

1. INTRODUCTION

Software Defined Networking (SDN) is a new paradigm of networking whose key tenets are: separation of data and control planes; a vendor agnostic interface such as OpenFlow [3] between the two; and a set of abstractions provided by the control plane such as a global network view, virtual network, or logical crossbar to make it easier to program and manage the networks. Transitioning the widely deployed Internet infrastructure to SDN represents a significant challenge and requires creative ideas for incremental deployment of SDN networks. During the transition, SDN networks need to coexist with traditional IP networks and any SDN deployment must be able to exchange reachability information, forward traffic, and express routing policies with existing IP networks. In this paper, we propose a solution for incremental deployment of SDN networks and how they can seamlessly peer with IP networks. The solution has been implemented and evaluated for feasibility and performance.

2. SDN-IP NETWORK PEERING

Peering between ASes (Autonomous Systems) on the Internet today is universally done with Border Gateway Protocol version 4 (BGPv4) [6]. Therefore we need a mechanism for an SDN AS to communicate with IP ASes via BGP.

The overall architecture is shown in Figure 1. This figure shows an SDN AS with gateway OpenFlow switches peering with external border routers from other ASes. To achieve peering, the centralized SDN control plane integrates a BGP process for the SDN AS. The BGP process handles routing updates with BGP peers on external IP networks. The entire SDN AS appears as a single router to its peers.

There is an SDN-IP peering application (including a BGP Route module and a Proactive Flow Installer) running on top of the

network operating system (NOS). The BGP route module is responsible for synchronizing BGP route updates pushed by the BGP process and storing them in a local RIB (Route Information Base). The Proactive Flow Installer uses the routes learned through BGP to calculate and install the flow entries for Inter-AS traffic.
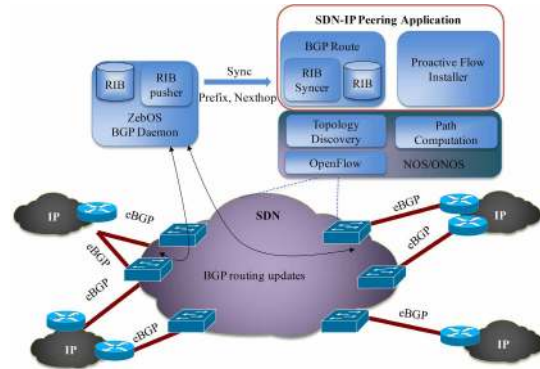

Figure 1. Architecture of SDN-IP Network Peering

2.1 RIB Synchronization

RIB updates from the BGP process are sent to the SDN-IP peering application running on the SDN NOS. We designed a RIB synchronization protocol for this purpose. This protocol is based on a REST API with GET, POST, and DELETE methods to sync the RIB, and additional APIs to coordinate behavior after process restart (Table 1). The API allows batching of RIB updates to send multiple updates in a single POST request. We plan to re-implement the protocol using a connection-based mechanism in the future, as we reach the speed, reliability and scaling limits of the current protocol.

Table 1. REST API for RIB Synchronization

Function	REST API
Initialization / Restart	/wm/bgp/<router-id>/<capability>
Single update	/wm/bgp/<router-id>/<prefix>/<nexthop>
Batched updates	/wm/bgp/<router-id>/rib/json

2.2 Proactive Flow Installer

The proactive flow installer in the SDN-IP peering application is used to convert each route in the RIB to flow entries in the FIB (Forwarding Information Base), and to proactively calculate and install the flow entries onto OpenFlow switches in the SDN AS. This action is triggered by BGP updates.

We use a concept of SDN edge and core to reduce churn on the OpenFlow switches when BGP routes flap. We do prefix based lookup at the first hop switch and MAC based forwarding in the SDN core (Figure 2). We proactively install a full-mesh of flow

paths between all pairs of gateway OpenFlow switches based on the gateways' MAC addresses. These flow paths are updated only when the internal SDN topology changes. At the first-hop switch, flow entries match external IP prefixes, and send packets on the appropriate flow path to the nexthop router learned from the BGP nexthop. When the nexthop for an IP prefix changes, the only thing that needs to be updated is the flow entry in the first hop switch. These flow entries update at the frequency of BGP route updates.
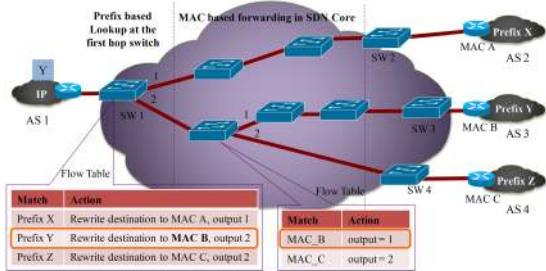

Figure 2. MAC Address Rewriting and Layer 2 Forwarding

2.3 Discussion on BGP Speaker

We chose a logically centralized BGP process for simplicity and correctness. Since all the external paths are selected by BGP, we are guaranteed not to break BGP semantics or create routing loops because BGP is computing the best paths. Additional benefits are centralized monitoring and software upgrade.

The BGP process needs to scale with number of peers and size of the RIB. In public IP peering use cases, a single BGP process can handle the full Internet routing table and the number of BGP peers. In service provider VPN (Virtual Private Networks) use cases, multiple BGP processes may be needed to handle the higher peer scaling. Our demo implementation uses a single BGP process.

There are two ways of getting BGP packets from the gateway OpenFlow switches to the BGP process. One approach is to provision BGP flows from all gateway switches to a server or VM (Virtual Machine) running the BGP process. Another is to intercept BGP packets at the gateway switch and shunt them to the BGP process through the control channel. The second approach is easier to provision but can stress the slow control path of OpenFlow switches. This demo uses the first approach and the deployment (in Section 5) uses the second approach.

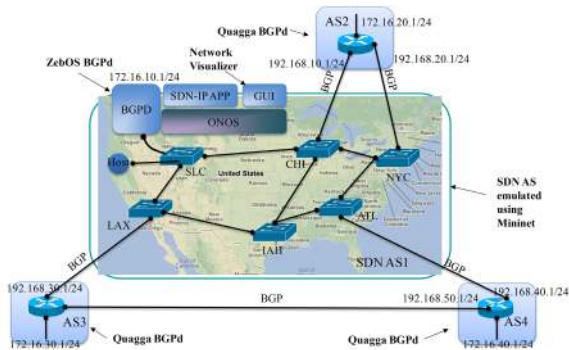## 3. DEMO IMPLEMENTATION

Figure 3. Demo Scenario and Configuration

The BGP process is ZebOS BGPd [5] modified to send RIB updates to the SDN-IP peering application. The SDN AS is controlled by ONOS (an experimental distributed SDN Operating System) [2]. We implemented SDN-IP network peering as an

application on ONOS. The demo network is emulated using Mininet [1]. Figure 3 shows the demo topology with SDN AS1 peering with three IP ASes.

Functional test: We created an inter-AS flow from a host in SDN AS1 to AS2, and a transit flow from AS3 to AS2. We demonstrate successful re-convergence when SDN internal links or SDN-IP external links fail.

Performance test: The SDN-IP peering application can scale to 10,000 RIB entries. The BGP process consumes 580 Mbytes of physical memory while the SDN-IP peering application (together with ONOS) consumes about 460 Mbytes (see Figure 4). It can process 100 RIB updates per second.
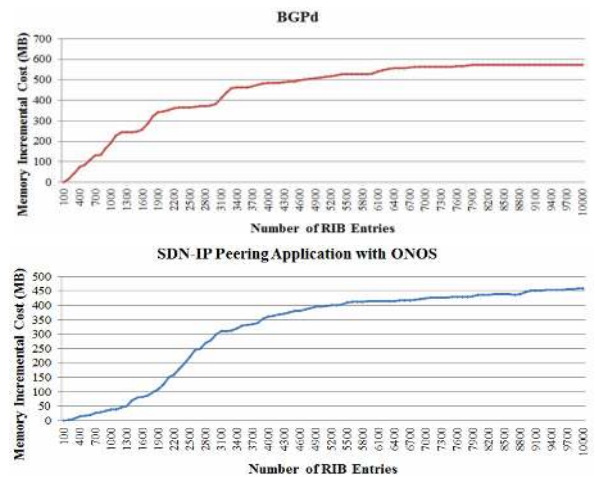

Figure 4. SDN-IP Network Peering System Performance after Inserting 10,000 Routes

4. RELATED WORK

RouteFlow [4] is one of the first implementations of IP routing on OpenFlow switches. RouteFlow instantiates a VM for each OpenFlow switch with as many virtual network interfaces as there are active ports in the corresponding device, and runs a stack of open-source routing protocols on the virtual topology. All control messages are exchanged between VMs as if they are running a distributed control plane. Such a solution incurs the overhead of distribution without the benefits of scale. The SDN-IP Peering Application is a simpler design, better integrated with SDN, and easier to implement advanced features such as traffic engineering and policy-based routing through the SDN domain.

5. FUTURE WORK

In collaboration with Google and REANNZ, we will deploy the SDN-IP network peering system to a small active SDN network in Wellington, New Zealand. Following this deployment we plan to improve the scalability and fault tolerance of the system.

6. REFERENCES
[1] Mininet: http://mininet.org.
[2] ONOS: http://onlab.us/tools.html.
[3] OpenFlow Specification.
    http://www.openflow.org/wp/documents/.
[4] RoutFlow: http://cpqd.github.io/RouteFlow/.
[5] ZebOS: http://www.ipinfusion.com/products/zebos-network-platform.
[6] BGP [RFC 4271]: http://www.ietf.org/rfc/rfc4271.txt.