Internetworking with SDN using existing BGP

Pingping Lin, Jun Bi, Hongyu Hu

Institute for Network Sciences and Cyberspace, Tsinghua University
Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
pingpinglin2012@gmail.com, junbi@tsinghua.edu.cn, huhongyu@cernet.edu.cn

ABSTRACT

The research on transiting the existing networks to SDN (Software defined networking) just starts. We propose to continue using the BGP (Border Gateway Protocol) and the legacy BGP border routers for the SDN network and connect the SDN network with legacy IP networks in this paper. The experiments we carried out preliminary proved our design.

Categories and Subject Descriptors

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

Keywords

Software Defined Networking; OpenFlow; BGP; Transition.

1. INTRODUCTION

At present, SDN is considered as a promising way to re-architect the Internet. Transitioning the existing network to SDN becomes an important issue to the network researchers in current stage. However, so far the SDN itself is not mature in many aspects and the research on the transition just starts. The main contributions of BTSDN are as follows: (1) It proposes that the controller runs Internal Border Gateway Protocol (IBGP) to learn the global inter-domain routing information. (2) It proposes a feasible transition solution for the existing networks to SDN through the existing BGP protocols and BGP border routers.

2. RELATED WORK

RouteFlow [1] is one of the implementations of IP routing on OpenFlow switches. The inter-domain uses the standard BGP. RouteFlow instantiates a VM (Virtual Machine) 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.

To enable incremental deployment of SDN, we conducted a research work of SND-IP network peering [2] in 2013. This work focuses on the interaction between BGP based SDN domain and legacy IP network. In this solution, we totally discarded the legacy border BGP routers in OpenFlow network. However, in the real world, border routers usually have a high performance and were bought with high prices. During the incremental deployment of OpenFlow, the network operators probably will not to throw the border routers. Thus, in this paper, we propose a more practical solution named BTSDN for transiting to SDN/OpenFlow network by retaining the border BGP routers.

3. ARCHITECTURE OF BTSDN

3.1 Overview of BTSDN

As mentioned above, BTSDN uses the BGP and retains the BGP routers in OpenFlow network. The geographical distribution of BGP border routers, OpenFlow switches, and controller is shown in the Figure 1. The inter-domain is still running the EBGP while the intra-domain is running the IBGP and OpenFlow protocol.
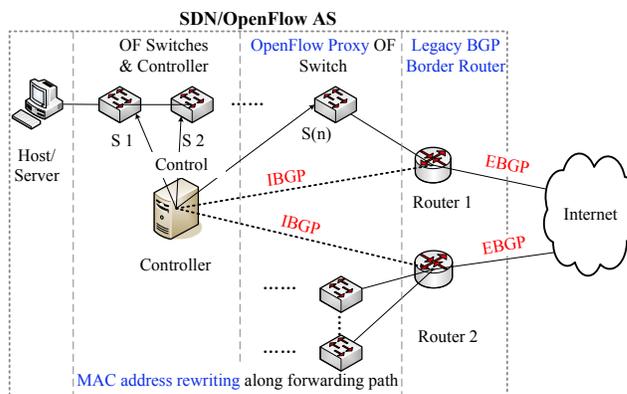

Figure 1. Architecture of BTSDN

3.2 Controller Running IBGP

By IBGP synchronization mechanism, an IBGP router can learn the global routing information in real-time. Therefore, BTSDN lets OpenFlow controller running the IBGP (we run a software BGP Quagga [3] as a network application above controller) to get the global routing information from the border EBGP routers.

3.3 OpenFlow Proxy and Flow Redirection

In BTSDN architecture, OpenFlow switch such as S(n) which is directly connected to a border router plays the role of the OpenFlow protocol proxy for the BGP border routers. Since the traditional BGP routers do not support the OpenFlow protocol, the controller

could not control the behavior of the border routers. But the controller can install certain instructions into the flowtable (specified in OpenFlow protocol) on OpenFlow proxy, and in this way indirectly controls the border router.

Once a data flow enters the OpenFlow domain, the first issue is how does a border router forward the first data packet to the downstream OpenFlow switch? This can be achieved by the redirect function in routers. The border router should redirect all the packets received from the Internet to the directly connected OpenFlow switch (OpenFlow protocol proxy). In this way the controller achieve the control to the entire intra-domain network.

3.4 ARP Proxy Module

We run software BGP software Quagga as a network application above controller, and let Quagga to talk to all the border routers. Before Quagga can talk to those routers, it needs to know the IP address and MAC address of each router. Thus, we configure the Quagga like configuring a normal router, and configure Quagga with all the boarder routers' IP addresses. Floodlight [4] can learn all the IP and MAC pairs of all the host/routers by its proxy ARP. So for the MAC addresses of those routers, we design and add an APR proxy module, which can extract the ARP entries from Floodlight and insert those entries into host's system ARP table.

3.5 Destination MAC Rewriting Module

Only when router/host/server receives a packet whose destination MAC is the same with the MAC address of corresponding router/host/server port, the router/host/server will accept the packet (except for special packets like broadcast packets). Thus, we add a destination MAC rewriting module to controller. It inserts destination MAC rewriting flow entry to switch. Before forwarding packet to router/host/server, OpenFlow switch will firstly resets the destination MAC address in the packet to the MAC address of router/host/server according to MAC rewriting flow entry, and then forwards packet to the router/host/server.

4. EXPERIMENT AND RESULTES

As shown in Figure 6, ten PC nodes emulate three ASes (Autonomous Systems). AS 1 is an OpenFlow AS. AS 2 and AS 3 are legacy IP (non-OpenFlow) domains. Three PCs (PC1, PC2, and PC3) play the role of web users and were installed with the Microsoft Windows XP Professional operating system. All other PCs including the controller are installed with Ubuntu-10.04-desktop-i386. The BGP software is quagga-0.99.20, and the switch software is Open vSwitch-1.4.0. Controller is Floodlight [4] and the FTP (File Transfer Protocol) server is WinFTP.

Step1: Verifying the feasibility of the controller running IBGP to get global routing information. Firstly, we use the static flow pusher API (Application Programming Interface) in Floodlight to install the BGP path between controller/Quagga and router 1. We start the BGP demon processes of Quagga software in the three border routers and the controller. Then router 2 announces IP address prefix 2.2.2.0/24 and 77.77.77.0/24, router 3 announces prefix 3.3.3.0/24, and router 1 announces prefix 1.1.1.0/24. Then, we look

up the route table in router 1, and find that router 1 has learned all the routes to the network 1.1.1.0/24, 77.77.77.0/24, 2.2.2.0/24 and 3.3.3.0/24. The controller also learned all the IP prefixes. Thus, the approach of letting controller running IBGP to gain the global routes is feasible.
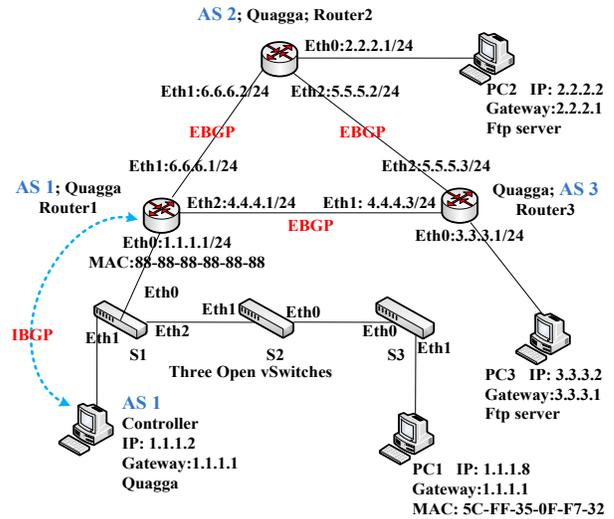

Figure 2. Experimental Topology

Step2: Verifying the feasibility of BTSDN solution. The target of this step is to verify whether PC 1 can communicate with PCs in other non-OpenFlow domains in proactive and passive models respectively. In this step, we firstly start the redirection function in router 1, and use the static flow pusher API of Floodlight to install the forwarding flow paths for PC1: (S1, S2, S3), (S3, S2, S1). We also add the destination MAC rewriting instructions into S3 and S1. After such configuration, flow path installation, destination MAC address rewriting, finally PC1 can upload and download files from all the FTP servers. The target of step 2 is achieved.

Evaluation: BTSDN uses BGP as Inter-domain routing protocol, so it inherits both the advantages and disadvantages of BGP. BTSDN does not adopt any new protocol or network equipment. It uses the traditional BGP and BGP border routers to transit the existing networks to SDN. BTSDN is a practical solution.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] RoutFlow: http://cpqd.github.io/RouteFlow/.

[2] Pingping Lin, Jonathan Hart, et al. "Seamless Interworking of SDN and IP", Proceedings of ACM SIGCOMM13.

[3] Quagga:http:// www.nongnu.org/quagga/.

[4] Floodlight:http://www.projectfloodlight.org/floodlight/.