Virtual Routing Engine: A Way of Migration towards SDN

Tianran Zhou, Delei Yu, Enhui Liu
Huawei

Introduction

By decoupling the network control from data forwarding, SDN (Software Defined Networking) and OpenFlow [1] explored a lot of novel capabilities and opportunities, which excites operators to deploy SDN. However, migration towards SDN will not take place at once, since legacy devices are largely deployed in the existing network. Operators may incrementally add new SDN devices in the legacy network or build a new software defined network in parallel, and manage hybrid networks. SDN devices have to cooperate with legacy devices by supporting traditional protocols, such as IGP and BGP.

Routeflow [2] is a platform for providing virtual IP routing services in OpenFlow networks. Each virtual router abstraction associates with one flow forwarding device. Cardigan [3] extended RouteFlow and abstracted multiple datapaths as a single router. Panopticon [4] presented the approach for aiding operators in planning and operating transitional networks.

In this paper, we propose a novel Virtual Routing Engine (VRE) system enabling the on-demand routing service and flexible operation for SDN. A VRE prototype is implemented based on OpenFlow 1.3 protocol. We deployed the prototype system in an 8 node hybrid network, and demonstrated the routing service with BGP and OSPF configurations.

VRE System Design

The VRE system architecture is shown in Fig. 1. The topology discovery module in the SDN controller detects OpenFlow devices, and monitors the states of nodes and links in the network. VRE server enables flexible topology abstraction of the physical devices and manages VRE instances. Each VRE instance is a virtual machine (VM) equipped with Linux based IP routing engine (e.g. Quagga). It runs legacy protocols and calculates routes for each virtual router abstraction. By integrating routes from VRE instance and the optimal internal paths of the virtual router abstraction, the VRE server generates holistic flow tables for flow forwarding devices. GUI and RESTful northbound interfaces are provided for flexible operation and on demand service provisioning.

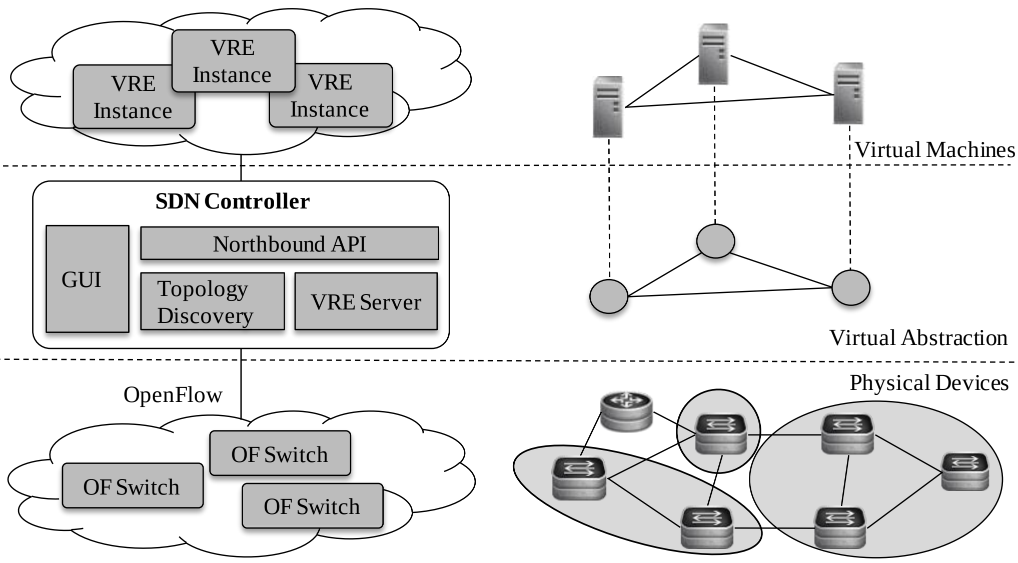

Fig. 1 Virtual Routing Engine System

The VRE system design consists of the following four novel techniques.

A. Flexible Topology Abstraction

The VRE system allows flexible topology abstraction, and dynamically maps VRE instances to flow forwarding devices to satisfy various requirements. For the incrementally added SDN devices in legacy networks, VRE instances can associate with them in a 1:1 configuration. For an entire newly built network or a cluster of adjacent SDN devices, they can be abstracted as a single virtual router. In a multi-tenant case, the OpenFlow network is sliced or virtualized into multiple ones, then a m:n configuration should be applied. The VRE server detects the topology boundary, rearranges the logical ports, calculates optimal internal paths and generates flow tables.

B. Automatic Path Recalculation

The VRE server subscribes events on topology changing (e.g. node or link failure) from the topology discovery module, and recalculates the optimal internal paths of a virtual router abstraction in real-time. Routes adjustment from the VRE instance can also be synchronized to the VRE server in time. So whenever network states changes, the VRE server automatically recalculates new paths and updates flow tables for packet forwarding.

C. Topology Information Assistant Optimization

With the knowledge of topology information, the VRE system intelligently detects connections among adjacent VRE instances, and sets up related logical paths. So the protocol packets can be forwarded directly among VRE instance VMs without going down to the switches. This optimization can reduce the latency of protocol packets transmission.

D. Routing as a Service

The VRE system implements the concept of routing as a service (RaaS), and exposes the capabilities, such as topology abstraction, VRE instance management and routing information inquiry, by a set of RESTful northbound API. With the help of the GUI, users can dynamically create a VRE instance mapping to one or more switches, configure the VRE instance with IP protocols, check routing and path in forwarding plane, and so on.

Demonstration

To verify the VRE system concept, we implemented the prototype with openflow 1.3, and set up a test bed of a hybrid network including six OpenFlow switches and two legacy routers. We demonstrated the routing service with BGP and OSPF configurations, and path recalculation.

In this preliminary experiment, a VRE instance can boot up within several seconds, including flow installation.

In the next step, we are going to put the VRE instances in a cloud environment and deploy the VRE system for geographically distributed networks.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, et al, “Openflow: Enabling Innovation in Campus Networks,” SIGCOMM CCR, vol. 38, pp. 69-74, March 2008.

[2] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, et al, “Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking,” HotSDN’12 Workshop, Helsinki, pp. 13-18, Aug. 2012.

[3] J. P. Stringer, Q. Fu, C. Lorier, et al, “Cardigan: Deploying a Distributed Routing Fabric,” HotSDN’13 Workshop, HongKong, pp. 169-170, Aug. 2013.

[4] D. Levin, M. Canini, S. Schmid, A. Feldmann, “Incremental SDN Deployment in Enterprise Networks,” SIGCOMM’13, HongKong, pp. 473-474, Aug. 2013.