SDN_CDN Documentation

Release 0.1.1

introom9

June 02, 2016

Contents

CHAPTER 1

What’s it about

SDN_CDN ain’t a pony.

```
       .,,.
     ,;;*;;;;,
    .-'``;-');;.
   /'  .-.  /*;;
 .'    \d    \;;               .;;;,
/ o      `    \;    ,__.     ,;*;;;*;,
\__, _.__,'   \_.-') __)--.;;;;;*;;;;,
 `""`;;;\       /-')_) __)  `\' ';;;;;;
    ;*;;;        -') `)_)  |\ |  ;;;;*;
    ;;;;|        `---`    O | | ;;*;;;
    *;*;\|                 O  / ;;;;;*
   ;;;;;/|    .-------\      / ;*;;;;;
  ;;;*;/ \    |        '.   (`. ;;;*;;;
  ;;;;;'. ;   |          )   \ | ;;;;;;
  ,;*;;;;\/   |.        /   /` | ';;;*;
   ;;;;;;/    |/       /   /__/   ';;;
   '*jgs/     |       /    |      ;*;
```

It is an ongoing experiment whose main focus is to make SDN better faciliate the performance of CDN.

CHAPTER 2

Get the code

You can checkout the source code here.

CHAPTER 3

Contents:

3.1 Overview

3.1.1 Components

Testbed

Mininet Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.

Switch

OpenVSwitch Open vSwitch is a production quality, multilayer virtual switch. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, 802.1ag).

Packet Analyzer

sFlow-RT sFlow-RT™ incorporates asynchronous sFlow analytics engine, delivering real-time visibility in Software Defined Networking (SDN) stacks and enabling new classes of performance aware SDN application such as load balancing and DDoS protection.

Flow Controller

Floodlight Floodlight Is an Open SDN Controller. Floodlight is designed to work with the growing number of switches, routers, virtual switches, and access points that support the OpenFlow standard.

Message Broker

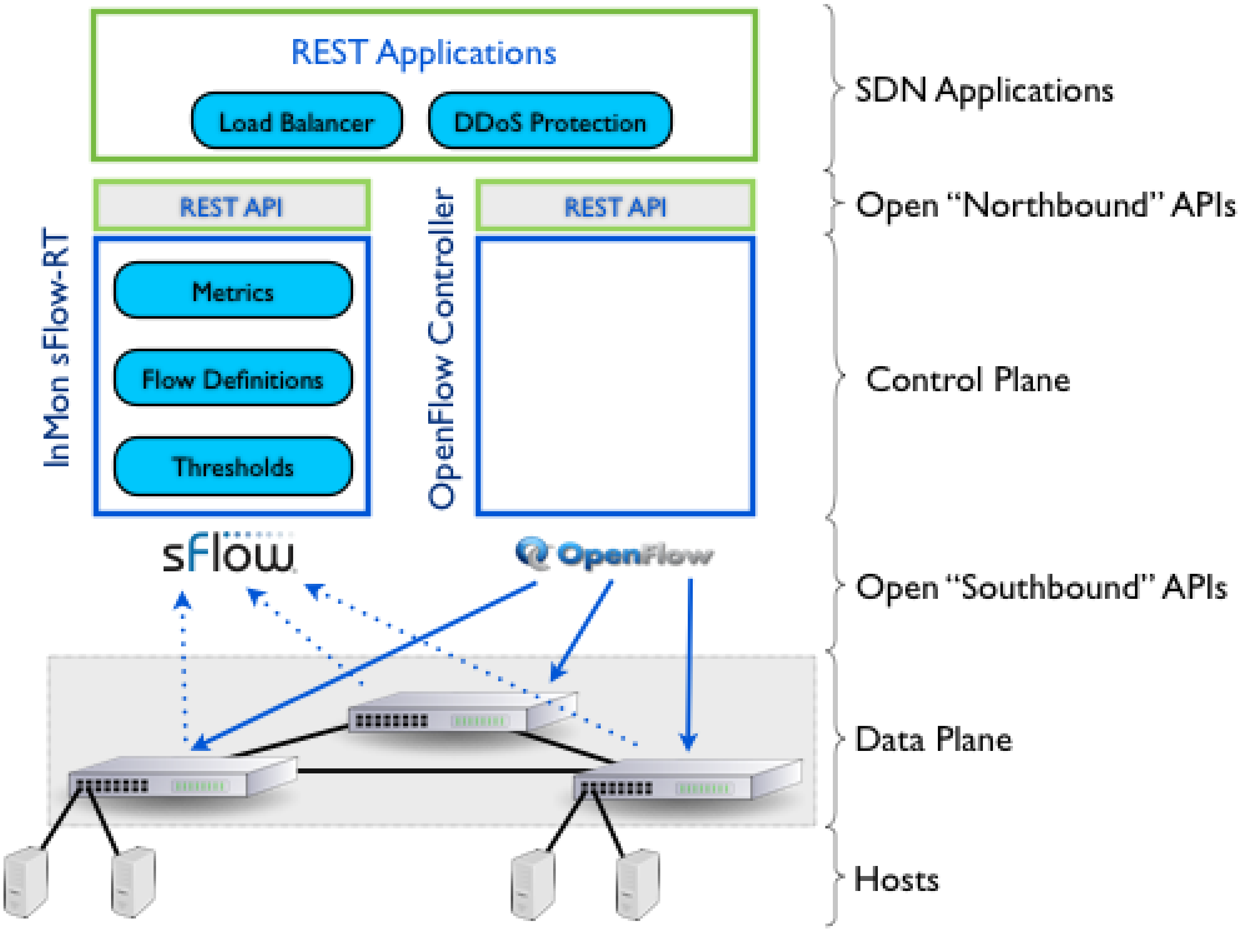
BestBroker BestBroker is a homebrewed meassage broker. It's job is to retrieve message from sFlow-RT and then provide well-formatted information to 3rd party.

DNS Server

SmartDNS SmartDNS is a homebrewed dns server. It will take into account information provided by BestBroker and return the best(least loaded) server address.

3.1.2 Architecture

The architecture is similar to the following graph.



Southbound

For the southbound, Mininet simulates a running network with customed topology and link status. The hosts/links are simulated directly by Mininet. The switches in the graph are powered by OpenVSwitch.

Northbound

For the northbound, Floodlight will act as an OpenFlow controller. Connections will be set between OpenVSwitches and Floodlight for the purpose of flow control/update. Meanwhile, sFlow-RT will act as flow/packet analyzer. The main job of sFlow-RT in this experiment is to analyze the traffic on each OpenVSwitch port. Sampled packet data will be transmitted from the switches of the southbound to sFlow-RT.

Note: BestBroker and SmartDNS are not pictured in the above graph.

Application Layer

For the application layer, BestBroker will invoke the REST API of sFlow-RT and then provide truned information to SmartDNS. Thanks to BestBroker, SmartDNS can select the best(least loaded) server as the DNS query answer.

3.2 sdn_module

Note: This page only gives the main ideas. Detailed information like how to install is too trivial to point.

SDN_MODULE is responsible for the northbound and southbound parts noted in the **overview** chapter. As you may have guessed, this module conatains both southbound and northbound. We will introduce them one by one.

3.2.1 Southbound

Launch:

To start **MiniNet** with a sample topology, you can invoke:

```
./sh/mininet_sample.sh
```

To set sflow sampling on **OpenVSwitch**, you can invoke:

```
./sh/ovs_set.sh
```

This will give us a running network.

3.2.2 Northbound

Launch:

To start **sFlow-RT**, you can invoke:

```
./sh/sflow_rt.sh
```

To start **OpenFlowController**, you can invoke:

```
./sh/controller.sh
```

Simulate Networking

During development, complex networking simulation is not quite needed. In this part, we will make use of the “iperf test” as an example. Login in to the Mininet interactive console. Then set hostX as iperf server and hostY as iperf client. More detailes can found in *./sh/reference/*.

3.3 cdn_module

3.4 roadmap