Controller Evaluation

Vivek Thejasvi Subramanyam, Praveen Kumar Kumsagi

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

Introduction

SDN was that it consists separate Control and Data plane. Data plane consists of all the forwarding devices namely switches, routers, etc. which are basically responsible for forwarding the packets from taking instructions and performing the action. Control plane consists of a large centralized controller because of which it will have better view of the network graph and makes better decisions.

Controllers

- Controllers to Networks is similar as Operating System to Computers
- Strategic Control point of the Network
- Relay Information to Switches, Routers, etc. through Southbound API's.
- Applications communicate through Northbound API's.
- Consists of number of pluggable modules that perform different tasks of the Controller.

- Most of the Controllers are Academic Projects by Students and Researchers
  - Designed to be Developer Friendly
  - All Users are students and Researchers
  - Less Concern towards Performance
- OpenDaylight and ONOS Controller
  - Industry grade projects focused for Commercial Deployment
  - Balance between Performance, Features and Developer Friendly
- Our Project Concentrates on OpenDaylight and ONOS

Architectural Components

OSGI

- Modularity Framework scheme for Java Development
- Without OSGI
  - As the Project grows, becomes more complex
  - Difficult to Understand
  - Difficult to Maintain and Update
- With OSGI
  - Modular Boundaries between bundles
  - Easier updates for the system
  - Improves Software Re-usability
- OpenDaylight and ONOS Controllers use OSGI Framework

Mininet

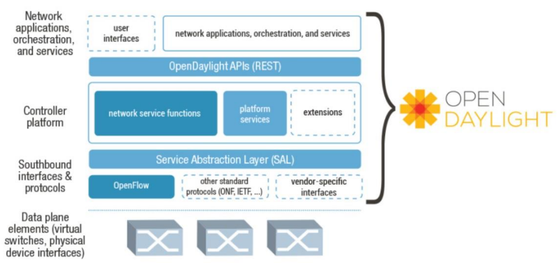
- Mininet is open source software which emulates the network
- Built-in Open vSwitch, and Open Flow capable switch
- Create Network through Built-in Scripts of by using Python Mininet Libraries.
- Equipped with Command Line and Set of very Useful Commands
- Used Mainly for Research and Development Purposes.

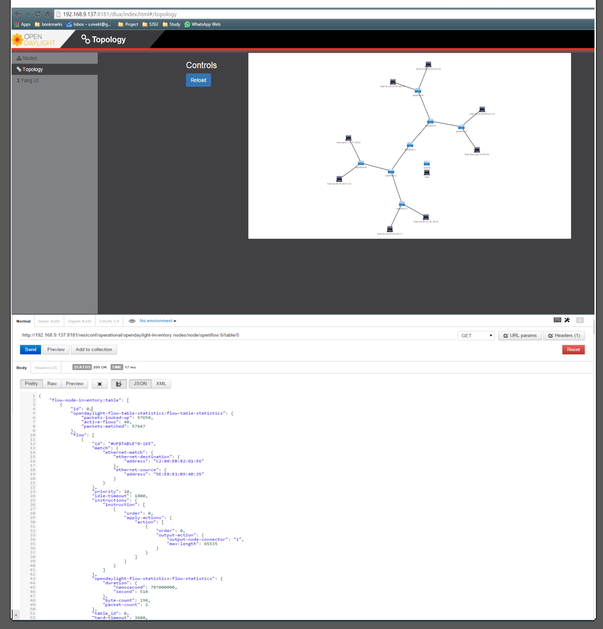
Apache Karaf

- It supports hot deployment of bundles during runtime
- It provides a console similar to Unix through which it can be completely managed
- It provides a rich set of commands that helps in management of configuration files. The changes to the configuration files are applied on the go.
- It supports a centralized logging system.
- Apache Karaf supports SSHD server, which allows you to connect to the console remotely.
- It provides a concept of 'karaf feature' which provides a way to describe and install and manage your applications.

OpenDaylight

- Opendaylight is an Open Source SDN software project under the Linux Foundation
- To build a strong and sturdy Open Source code base that covers all of the major components that is required for building an SDN solution
- To build a controller that will attain industry wide acceptance among the vendors and users
- To build a thriving and growing community around the Open Source code base that will keep adding value above, below and around the existing code.
- Main Contributors: Brocade, Cisco, Citrix, Ericsson, IBM, Juniper Networks, Microsoft and RedHat





Southbound interfaces and protocols consists of the protocols that the controller uses to talk to the network devices. All the southbound protocols are added as plugins in the Service abstraction layer. SAL figures out how to perform a requested service by the above layers on the network devices irrespective of the underlying protocols between the controller and the network devices. Each plugin is designed to work independent of each other. As new protocols come up in the industry, It can be included in the controller as a plugin in SAL.

The controller covers most of the basic functionality of the controller. The Topology manager builds the topology of the underlying network topology by using the help of other components like ARP handler, Host Tracker, Device Manager and Switch Manager.

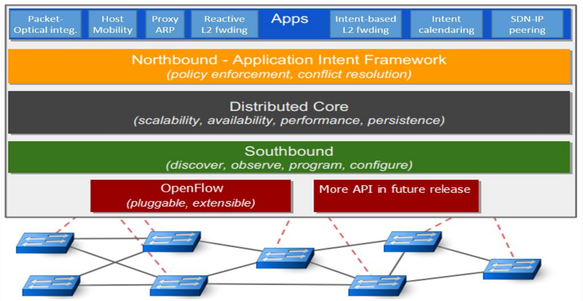
ONOS

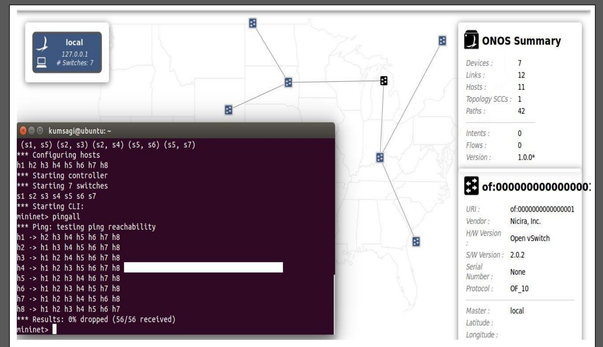
- ONOS is the first open source network operating system specifically target at service providers and critical mission networks.
- ONOS is purpose built to provide the high availability (HA), scale-out, and performance these networks demand.
- In addition, ONOS has created useful Northbound abstractions and APIs to enable easier application development and Southbound abstractions and interfaces to allow for control of OpenFlow-ready and legacy devices

At the heart of the Onos is the distributed core which is responsible for providing scalability, security, services, accepting sensory information from southbound API and distributing them among the controller clusters.

To the South side of the architecture is a boundary which is referred to as Southbound API and this Southbound API is primarily responsible for acting as a barrier and limiting protocol specifics by entering into the distributed core and further penetrating into Northbound APIs. So basically Onos has a very strong firewall features.

To the North side of architecture it has Northbound API which possess applications pertaining to certain features. Northbound abstraction basically consists of low level abstractions like devices, switches, etc. and these low abstraction together form the network graph. On top this low level abstraction there is Application Intent framework which has goal oriented approach to the network.





Project Accomplishment

- Researched and Studied Architecture of OpenDaylight and ONOS
- Successfully Documented Installation and Running OpenDaylight and ONOS
- Researched on Additional Features like Flows, etc.
- Created Scripts on Installing requires versions of Dependencies, Downloading and Building the Controller.
- Issues

Conclusion

- OpenDayLight and ONOS Controllers are designed keeping modularity, future protocols, and updates in mind
- These controllers are not only built as an industry grade controllers but also to be easy to develop applications on
- They are built to support multiple southbound and northbound protocols and also keeping future changes and new protocols in mind
- These Controllers are bit difficult to understand and start with due to their complex Architecture as compared with its counterparts.

References

- VOLKAN YAZICI http://vlkan.com/blog/post/2013/07/31/openflow-controllers/
- POX Wiki - https://openflow.stanford.edu/display/ONL/POX+Wiki
- Murphy McCauley - http://www.noxrepo.org/
- http://karaf.apache.org/
- https://**maven**.apache.org/
- http://karaf.apache.org/
- https://**maven**.apache.org/

Acknowledgement

We would like to thank our project advisor, Dr. Greg Bernstein for all the help and guidance that he provided us for implementing and completing the project successfully. We would also like to thank Santosh Jayashankar for his work on ONOS controller.