Floodlight Controller onto Load Balancing of SDN Management

Mohammad Qassim¹, Mohammed Najm Abdullah², Abeer Tariq³

Dept. of Comp.Science, University of Technology

Abstract

For managing SDN are still working, but some things have become quite clear. Network management systems will need new interfaces for interacting with cloud orchestration systems and SDN controllers. New network devices that provide a great deal of data about the overall network, SDN controllers also have tremendous impact on network state, given their ability to program other network devices. Two-way integration between a network management system and an SDN controller allows that network management system to do the following. Despite these regular improvements and upgrading, computer networks of today are struggling to meet the heterogeneous and ubiquitous society’s demand. The architectural inflexibility of computer networks gives researchers challenges in experimenting new ideas. The paradigm SDN then ONF architecture, open ways for implementation of a programmable network architecture to be applied gradually in production networks, software switches into a virtual network environment replace physical switches. Software Defined Networking into Graphic User Interface, Floodlight controller was implemented as the network OS. Adapting its codes to develop components; packet forwarding/address learning, load balancer and firewall for optimizing network. This research achieved a state of the art network management from both a centralized after that Graphic User Interface...

Keywords: SDN, Floodlight, OpenFlow, Mininet

1. Introduction

Continually states of network changes, and network operators must manually adjust network configurations in reaction to dynamic conditions of the network [1]. External tools are frequently used by network operators to manage their network, some end up building adhoc scripts to dynamically reconfigure devices on the network to handle events as they occur resulting in frequent changes and misconfigurations [2]. A new paradigm Software Defined Networking (SDN) has presented an alternative to manage such limitations. SDN which implies that the switch is confined to a simple packet-forwarding device. A software-based controller now manages a principles followed via a switch onto the forward packets to destination [3]. The applications can now control the switches by running on top of a network operating system. OpenFlow [4] [5] [6] [7] [8] was introduced onto the standardize the communication during switch (data plane) then the software-based controller (control plane)
into SDN OpenFlow networks possess certain capabilities like control of multiple switches from a single controller, analysis of traffic statistics using software, dynamic update of forwarding information and so much more [3]. There are several SDN controllers available, examples are NOX [9], Beacon [10], Maestro [11], Floodlight [12], Frenetic [13], Trema [14], and etc. The programming language used in coding a controller will be same as a programmer will use to code applications for that controller. Floodlight is an SDN-OpenFlow controller that comes with built-in applications [12]. A set of common functionalities be utilized by Floodlight Controller onto probe and control the OpenFlow network, when implementations onto the top achieve various characteristics into obtain resolve various employer requires over the network. Floodlight architecture, consists of the 3 inter-connected parts from the Floodlight Controller, implementations create Java modules compiled within the Floodlight, then implementations built over the Floodlight REST API [15]. Java be utilized onto increases network running modules then in compiled them with floodlight controller module uploading system. Mininet be system which t supplies huge networks from the fast prototyping onto the computer. Utilizing the lightweight virtualization techniques that makes a scalable SDN, like the processes, network namespaces. Employers be make, interact within to be faster utilizing the mininet.

2. Styling then Methodology

Codes for these functional components/modules were done in Java, based on Floodlight Controller and then implemented in Mininet simulated network. The network be making styling onto run on the computer running a SDN FloodLight, 7 switches and 8 hosts. This architecture replicate data center architecture within three depths the  S1, the switches S2 and S5 then The Edge Switches S3, S4, S6 and S7, with each edge switch connecting 2 hosts.

create the network a guest machine was setup on V.Box “Software Defined Networking within a different controllers” managing into the Ubuntu 14.04 64 bit O.S. 2GB within multiple processor, then setting 2 adapter interfaces from the NAT.
Inside the guest machine, the **Mininet** installed to emulate a network the utilize as checked, **ONF or ONF-V switch** the **Floodlight Eclipse** that came with the Operating System which will be very essential, for example Terminals, xterm, Emacs, browser etc. was setup Setting up to run Floodlight, Virtual machine powered up and a new workspace maked into eclipse.

![Fig 2: V.B.running my SDN with Floodlight controllers](image1)

The code given in Table 1 below was entered in a Terminal to create and connect the network to Floodlight controller.

```bash
$ sudo apt-get update
$ wget clone http://www.mininet.com/mininet/
$ sudo mn controller=remote,ip=127.0.0.1.port=6653 mac topo=tree,3 switch ovsk,protocols=OpenFlow13
```

![Fig 3: Floodlight loaded on eclipse](image2)
Immediately the network on Mininet connects to floodlight controller running on eclipse, a handshake (HELLO PACKET sent and received) occurs to confirm the connection.

3. Outcomes within :

A- **Pingall command:** by using this command will making all network hosts transmit ping packets onto other in order to confirm the communication between them then the reachability. making utilizing the Internet Control Message Protocol.

![Fig 4: Pingall command triggering packet forwarding amidst all hosts into network](image)

the tools accomplishing an aim of the purpose of pinged each one another. The address learning module the flow table within together a MAC and IP addresses to each destination.

![Fig 5: GUI display of all network devices](image)
B. iperf command onto terminal utilizing

above graph exhibits TCP bandwidth amidst hosts into the network which be utilizing Floodlight controller within 5 sec:

- Switch 3, h1 and h2: data was transferred at a rate of 9.67 Gigabits per second
- Switch 4, h3 and h4: the rate at which data was transferred 9.3 Gigabit per second
- Switch 6, h5 and h6: data transfer rate was 9.39 Gigabits per second
- Switch 7, h7 and h8: the data transfer rate was 10.0 Gigabits per second

the check amidst hosts, be terminated which the Address learning combination executed onto a switch flow schedule within the addresses releated onto port development data transfer onto the origin within destination switch.

Fig 6: Bandwidth between hosts using Floodlight controller.

Fig 7: Ping from hosts H1 to H7 confirming communication deny by firewall.
Fig 8: Ping command from H1 to H7 confirming communication been allowed by firewall

C. executing the LOAD BALANCER combination

within the load balancing while firewall be running the alittle part from a principles previously inserted, no other principle configured the firewall.

Entering the command below into a terminal will disable firewall. simple load balancer application module be executed into ping, tcp, udp services. ping service be utilizing now Load balancing be making a straightforward intelligent routing obviate a overcrowding as well as to duly utilize network resources. Load balancer works with set of criteria, customer requesting the service, the service, the pool, the virtual server.

A system during the REST API query attached devices onto Floodlight be displayed when exists. also, routing service capable of evaluate route from a load-balanced packets traversing ..

Load Balancer module will load balance the ICMP packets during hosts 3(10.0.0.3) then host 4 (10.0.0.4) (BTW "port":"8" signifies ICMP packets internal to the LoadBalancer module):

```bash
curl -X POST -d '{"id":"1","name":"pool1","protocol":"icmp","vip_id":"1"}’ http://localhost:8080/quantum/v1.0/pools/
curl -X POST -d '{"id":"1","address":"10.0.0.3","port":"8","pool_id":"1"}’ http://localhost:8080/quantum/v1.0/members/
curl -X POST -d '{"id":"2","address":"10.0.0.4","port":"8","pool_id":"1"}’ http://localhost:8080/quantum/v1.0/members/
```

builds a server within IP address 10.0.0.100 after that making a pool and add members (10.0.0.3 and 10.0.0.4) of the network to the pool as servers that will render the service needed by the requesting client. After the load balancer is configured, h1 pinged 10.0.0.100 (PS: 10.0.0.100 does not exist in our network,
initial time delay be while a LoadBalancer module integrated necessary flows from switch along a path (10.0.0.3). Flows integrated via LoadBalancer module be making see via demanding flows from switch in a terminal widow.

Fig 10: the Ping the host H2 to virtual server 10.0.0.100.

a result from execution, messages interchange via switch then OpenFlow protocol onto Software Defined Networking architecture noticed onto obviously grasp to know how makes doing after that know how utilizing this from the control. be displayed which a real environment be not a must, to commence design to make a prototype implementations, virtual networks be helpful, beneficial from coding, testing then stratify modern implementations.

4- Conclusion

Network disposition, administration be completely an overawe duty, hardness from today network architecture make subjoin onto a misfortune network factors interfaces. tell by the amount of industrial buy-in SDN has seen by networking corporations like Cisco, HP, etc, developing SDN capable switches for programmable networking architecture, be making the Software Defined Networking, Open Flow as a standard. Opened up the
Mininet emulator then show onto their applications in a virtual scenario for progress deployment a networks. a Controller as main part from Software Defined Networking which be making a discuss. Enforcement scenario, a floodlight controller in eclipse be performed then will showed the network into Graphic User Interface. , which be making Graphic User Interface networking execution. result from execution be analyzed onto display connection via hosts.

Software Defined Networking as yet developing, so several territories as yet within require from research be capable of onto create network to be make a full interact within innovate onto in a next generation. intend onto expand a work onto future to see how more networks be making communicated onto other for routing component, make a connection between hosts from a various networks.

Reference