Software Defined Networking

By

Steven Couture

A Proposal Submitted to
The Faculty of the Department of Information Technology
In Partial Fulfillment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

University of Cincinnati
Department of Information Technology
College of Education, Criminal Justice, and Human Services

April 2014

Learner’s signature: ___________________________ Date: 4/20/15

Learner’s printed name: Steven Couture

Advisor’s signature: ___________________________ Date: 4/20/15

Advisor’s printed name: Professor Mark Stockman
Table of Contents

Section

I. Abstract  pg. 1
II. SDN Description  pg. 2
III. Project Description  pg. 3
IV. OpenDaylight  pg. 3
V. Budget  pg. 4
VI. Problems Experienced  pg. 5
VII. User Profile  pg. 6
VIII. Project Timeline  pg. 7
IX. Network Topology  pg. 7
X. Conclusion  pg. 9
XI. Works Cited  pg. 9

Figures and Tables

Figure 1: Traditional vs. SDN Architecture  pg. 2
Figure 2: OpenDaylight Build  pg. 4
Table 1: Project Timeline  pg. 7
Figure 3: Project Topology  pg. 7
I. Abstract

Software Defined Networking (SDN) has the potential to completely change the way IT manages their networks. Networks are becoming increasingly complex; SDN can allow a network operator to strip the intelligence off of hardware devices such as switches and centralize them on a controller. This can provide many benefits to a networking environment by allowing IT to perform such acts as static routing, flow tables, and simply observing network traffic statistics from a single location which can in turn help reduce operating costs. SDN Market is predicted to reach $3.52 billion by 2018, growing at a rate for 61.5 percent between 2012 and 2018 (Burt). Well-known networking companies such as Cisco, Juniper, IBM and Hewlett-Packard as well as Big Switch Networks, Intel, VMware, and Google are major contributors in the open source SDN market as well. I will explore just what exactly is SDN, what abilities it gives you from a networking standpoint, and why this technology is taking off.
II. SDN Description

Software Defined Networking differs from your traditional networking infrastructure due to the addition of a centralized controller. In an SDN environment all network changes can be made from this single controller location as opposed to your traditional networking of accessing each individual switch. This controller builds an end-to-end view of the network and separates a networks data planes and control planes from the network’s switches which in turn allows a user to manage the network from a single location. There network paths can be configured based on different sections of a packet header.

![Figure 1: Traditional vs. SDN Architecture](image)
III. Project Description

In this project I have built a small three switch, four host network using the network emulator Mininet and an Ubuntu virtual machine that is utilizing the SDN controller OpenDaylight to manage this network and demonstrate some of the abilities SDN can provide to that network. I have created a flow table using OpenDaylight’s web user interface that will redirect web traffic intended for host 1 to instead be sent to host 2 which will be hosting a simple web server. I will also compare and contrast some of the more prevalent SDN controllers as well as the status of SDN in technology now.

IV. OpenDaylight

OpenDaylight was the the SDN controller used in this project. OpenDaylight appears to the front runner for the preferred open source SDN controller in the market with support from companies including Cisco, Citrix, Dell, HP, IBM, Juniper, Microsoft, VMware and many more companies. OpenDaylight is a “community-driven, open source controller platform” and released their newest version of their controller called Helium just this past March 19th. However, I have opted to use the first release of ODL’s controller named Hydrogen based on the availability of documentation for this particular release.
V. Budget

Everything used in this project is free and open source (Virtual Box, Ubuntu, OpenDaylight, and Mininet). If SDN were to be implemented in a company’s network the price would vary based on the size of the network but would remain fairly cheap. Most SDN controllers are free and open source including OpenDaylight (which is demonstrated in this project), Floodlight, POX/NOX, and RYU. Any costs would simply be hardware for a server to host the SDN controller. For large networks they might want several controllers which might require several servers.
VI. Problems Experienced

Many of the problems experienced in this project are due to the fact that Software Defined Networking is a fairly new technology and much of the project had to be self-taught. For example the SDN controller used in the project, the network emulator, the concepts of SDN, and the coding languages used. For example, it is possible to write very complex flow table rules for the network based on a controller’s programming language. OpenDaylight uses Java; POX uses Python. Getting the controller to connect to the network emulator and build the correct topology in the web user interface has also been a challenge. There is also a bug in the controller’s web UI that disables the Flow tab where flow tables are created, deleted, edited, installed or uninstalled. Though the flow table remains intact and the network still behaves as it should there is no changing this through the web UI after the first flow is installed. It is still possible to make changes as a developer but because of the timing of the bug and not having a developer on the team it was decided to work through the bug.

Other challenges have been mostly hardware complications. Prior to the environment being moved to the IT Sandbox having the Mininet VM and it’s 8 hosts and 7 switches running along with the Ubuntu controller put a strain on the host system causing it to run at an incredibly slow speed. Once the environment was moved onto the Sandbox this issue was resolved. However, when attempting to expand the network from 8 hosts to 16 hosts the IT Sandbox struggled with all of these nodes. Network expansion had to be abandoned because of this complication and for the demonstration the network was scaled back even more to 4 hosts.
VII. User Profile

Potential users:

Any IT staff would be a user of their SDN network.

Software and Interface Experience:

Software experience would be useful but not entirely necessary as flow tables and management can be done through the web user interface for OpenDaylight. However, if more complex flows were desired in the network the software experience would depend on which SDN controller was being used which is Java for OpenDaylight. Of some of the more prominent SDN controllers, Python, Java, or C++ would be recommended.

Experience with Similar Applications

Most users will not have any experience with SDN controllers.

Task Experience

Setting up SDN on a network would require training or a firm grasp of the concepts based on the topic being fairly new. The user will need to have an understanding of their network.

Frequency of Use:

Daily; Depending on the type of network environment at each company, a network operator could change the flow of traffic based on time-of-day needs or unpredictable events such as DDOS attacks.

Key Interface Design Requirements that the Profile Suggests:

Configuring the SDN controller correctly (OpenDaylight) so that you can view the network topology from the web GUI to easily manage the network.
VIII. Project Timeline

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Mode</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Find a Topic</td>
<td>7 days</td>
<td>Mon 8/25/14</td>
<td>Tue 9/2/14</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Research</td>
<td>45 days</td>
<td>Mon 9/1/14</td>
<td>Fri 10/31/14</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>SDN Concept</td>
<td>45 days</td>
<td>Mon 9/1/14</td>
<td>Fri 10/31/14</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>SDN Technologies</td>
<td>45 days</td>
<td>Mon 9/1/14</td>
<td>Fri 10/31/14</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Network Planning</td>
<td>10 days</td>
<td>Mon 11/3/14</td>
<td>Fri 11/14/14</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Define Network Requirements</td>
<td>10 days</td>
<td>Mon 11/3/14</td>
<td>Fri 11/14/14</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Define Network Topology</td>
<td>10 days</td>
<td>Mon 11/3/14</td>
<td>Fri 11/14/14</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Decide SDN Technologies</td>
<td>10 days</td>
<td>Mon 11/3/14</td>
<td>Fri 11/14/14</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Implementation &amp; Testing</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Create Documentation</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Create Network</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Configure Network</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Configure SDN</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Test SDN Settings</td>
<td>61 days</td>
<td>Mon 11/17/14</td>
<td>Mon 2/9/15</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Presentation Preparation</td>
<td>36 days</td>
<td>Tue 10/15</td>
<td>Tue 10/15</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Create Diagram Display of Network</td>
<td>36 days</td>
<td>Tue 10/15</td>
<td>Tue 10/15</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>Prepare Talking Points</td>
<td>36 days</td>
<td>Tue 10/15</td>
<td>Tue 10/15</td>
</tr>
</tbody>
</table>

Table 1: Project Timeline

IX. Network Topology

```

[Diagram of network topology]
```

Figure 3
X. Conclusion

Software defined networking will change the way IT does networking in the coming years to help manage increasingly complex networks. Whether companies decide to go with an open source controller such as OpenDaylight or with a proprietary controller like Cisco’s SDN Controller, ONE, remains to be seen. Regardless the ability to strip the intelligence from networking devices and centralize them creates a major benefit in managing these complex networks and in return reducing operating costs. SDN is an up and coming technology that should be monitored closely by tech companies both large and small that are interested in keeping up with the ever evolving world of IT.

XI. Works Cited


"OpenDaylight Delivers Open Source Software to Enable Software-Defined Networking."


