SDN Demystified

Dean Pemberton – NSRC
Who am I

• Dean Pemberton
  – NSRC
    • Trainer/Network Engineer
  – Victoria University of Wellington
    • SDN Research Associate
You probably have questions

• What is SDN?
• What's wrong with the network I have now?
• What can an SDN do?
Software Defined Networking is...

• The stupidest name ever invented.
Software Defined Networking is…

• SDN allows network administrators to manage network services through abstraction of lower level functionality.

• This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane).
Software Defined Networking

• You’ve probably had Software Defined Networking for years?
• Anyone own a Juniper M-Series?
• It was just that you were never allowed to define or control the software.
Let's go back in time
Remember this…
Remember when…

• If the features you wanted were supplied by the operating system you were in luck.
• =)

• If the features you wanted were not supplied by the operating system, there were limited opportunities to expand it to include those features.
• =(  

UNIVERSITY OF OREGON
Enter choice
End User Innovation

• With Open Source Operating System Software control over the development and deployment of OS features is placed in the hands of the users.

• If you need a feature, even if you are the only one on the planet who wants it, you have a way to develop and deploy it.
A world without...

- Facebook
- Google
  - https://developers.google.com/open-source/
- Android
- etc.
Now think about current network equipment…

• Do we currently live in a world more like the closed source OS past?
• Or the current OS world where end users can innovate.
Current Network Feature Roadmap

• You have a good idea
• You go to your network vendor and pitch the idea
• Your network vendor asks how many units you’re going to buy
• That number is not enough
• Nothing happens regarding your good idea
Current Example

• “Hi Mr Load Balancing Vendor, I’m a ccTLD in a small country, we face a set of unique challenges with regard to managing bandwidth and protecting against DDoS attacks. We own 2 of your units and were wondering if you might be able to develop some features to assist us in these unique challenges”

• *CLICK* brrrrrrrrrrrrrrrrr
Another Example

• “We are pleased to announce that after months of development the new version of our networking software will support <feature X which you don’t need>. The price for the next software upgrade will be double to re-coup this development cost.”
What if we lived in a world where…

• You could start an open source project where people could develop the features you actually needed your platform to support.
• You didn’t need to pay for features that you were never going to use.
• You didn’t need to worry about bugs in code you were never going to use.
This works today for OSs

• If you need a new extension to Apache/BIND/MySQL/etc. then you can have someone develop them for you.

• What if you could do the same thing for all the features in your:
  – Switches
  – Routers
  – Load Balancers
  – Firewalls
Software Defined Networking

• Allows you to do just that.
• It allows you to take back control of the software that controls your network.
• It allows you to drive the speed and direction of the innovation of features within that software.
How?
Software defined networking (SDN)

• Separates control and data plane:
  – Open interface between control and data plane (OpenFlow)
  – Network control and management features in software
...SDN
Linton 3 Layer Model
Lessons from history 😊

• "If you know what you're doing, 3 layers is enough; if you don't, 17 layers won't help you.”

• [B]eware of the panacea peddlers: just because you wind up naked doesn't make you an emperor.
   – Michael A Padlipsky
Openflow overview

- One of the key technologies to realize SDN
- Open interface between control and data plane

![OpenFlow Diagram]

<table>
<thead>
<tr>
<th>match field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac dst</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Flow Table
1. Forward packet to on or more local ports
2. Encapsulate the packet and forward to controller
3. Drop the packet
4. Send through switch’s normal L2/L3 processing
## Flow Rule Examples

<table>
<thead>
<tr>
<th>Mask</th>
<th>Input Port</th>
<th>Source MAC</th>
<th>Dest MAC</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>Source IP@</th>
<th>Dest IP@</th>
<th>IP Proto</th>
<th>IP SrcPort</th>
<th>IP DstPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Switching</td>
<td>*</td>
<td>*</td>
<td>12:2E</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>IP Routing</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>App Firewall</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>443</td>
</tr>
<tr>
<td>Flow Switching</td>
<td>Port6</td>
<td>12:2E</td>
<td>17:FF</td>
<td>0800</td>
<td>VLAN7</td>
<td>1.2.3.4</td>
<td>4.3.2.1</td>
<td>06</td>
<td>11317</td>
<td>80</td>
</tr>
<tr>
<td>VLAN + App</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>VLAN7</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>80</td>
</tr>
<tr>
<td>Port + Ethernet + IP</td>
<td>Port6</td>
<td>12:2E</td>
<td>*</td>
<td>0800</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Examples

• Layer 2 – Switches
• Layer 3 – Routers
• Layer 4 – Load Balancers
• Layer 4+ - Firewalls
Layer 2 – Switches

• Network Virtualisation
• Data Centre
• Multi Tennant
• FlowVisor

• Each customer not only gets their own ‘network’ they can control it with their own controller.
Layer 3 – Routers

• RouteFlow
• What if you were able to take any number of ports throughout your network and draw them together into a router?
RouteFlow
Cardigan overview

- REANNZ AS 38299
- switch #1
  - BGP + traffic
- switch #2
  - BGP + traffic
  - traffic
- GLOBAL RPKI DB
- WIX AS 9439
- quagga + rtrlib controller
- Openflow + BGP
- CARDIGAN
Layer 3 – Routers

• Being able to add new features without waiting for vendor support
• RPKI
Layer 4 – Load Balancers

• Load Balancers need to take into account not only complex information about network latency, congestion and performance, but also the load on each of the servers that they are balancing traffic across.

• They also need to know how the balanced application deals with certain situations

• The best person to know that is YOU
Layer 4 – Load Balancers


Layer 4+ - Firewalls

• We install firewalls everywhere
• They are expensive
• What if we could somehow virtualise them and deploy them only where needed.
Layer 4+ - Firewalls


• Gamayunov, Dennis, Ivan Platonov, and Ruslan Smeliansky. "Toward Network Access Control With Software-Defined Networking."
Current Work in NZ on SDN

- Parallel REANNZ backbone
- VSD (Victoria Standard Distribution)
- RPKI on CARDIGAN
- NZIX2 at Citylink
- SDN being taught to undergrads in Q3/2014 at VUW
NZNOG SDN Install Tutorial

- SDN Intro
- Ryu – OpenFlow Controller
- Open vSwitch
- RouteFlow

- Building a L2 Switch
- Building a L3 Router
NZNOG SDN Install Tutorial
Takeaways

• SDN separates the control of the network from the elements involved in actually forwarding the packets
• This allows us to have a holistic view of the network not available before
• SDN allows you to control the direction and speed on innovation.
• Active area of development
• Watch this space
Questions

Do you have any questions?