15th Annual Global LambdaGrid Workshop
Plenary Sessions
Sep 30th

Running production and experimentation at AmLight SDN

Jeronimo Bezerra
Florida International University
<jab@amlight.net>
Outline

• Context
• Motivation
• Architecture
• Methodology
• Results
• Future
Context

*AmLight is a Distributed Academic Exchange Point*

- **Production** SDN Infrastructure (since Aug 2014)
- Connects AMPATH and SouthernLight GLIF GOLES
- Carries Academic and Non-Academic traffic
  - L2VPN, IPv4, IPv6, Multicast
- Supports Network Virtualization/Slicing
  - Openflow 1.0
  - Flow Space Firewall for Network Virtualization/Slicing
  - OESS for L2VPNs
  - NSI(OpenNSA+OESS) and OSCARS enabled
    - Including AMPATH and SouthernLight
  - Currently 4 slices for experimentation (including ONOS SDN-IP)
Context (2)
Examples – ONOS SDN-IP @ ONS
Examples (2) – ONOS SDN-IP @ ONS
Examples (3) – And more...

• In partnership with RNP:
  – FIBRE (*Future Internet testbeds / experimentation between BRazil and Europe*): how to use an OpenFlow native backbone to interconnect FIBRE islands (or racks)?
  – FIBRE island installed at AMPATH/Miami and using AmLight
• In partnership with Internet2:
  – Internet2 Technology Exchange 2014 – Multi Domain controller managing slices from different SDN domains (Internet2, AmLight, Univ. of Utah and MAX)
  – Internet2 Global Summit – ONOS SDN-IP demonstration
• In partnership with University of Twente:
  – “*Assessing the Quality of Flow Measurements from OpenFlow Devices*”
  – Authors: Luuk Hendriks, Ricardo de O. Schmidt, Ramin Sadre, Jeronimo A. Bezerra, and Aiko Pras
• All of them running on the same *production* infrastructure
Motivation

• How to guarantee experimental applications won’t affect my “production” slice?

• FlowSpace Firewall slices based on <switch, port, vlan>:
  – No extra filters are possible at this moment

• Multiple OF controllers could manage the same OpenFlow device:
  – Complicated to isolate who is sending specific OF messages

• OpenFlow deployed by some vendors is still “experimental”:
  – Unsupported messages could lead to a device crash

• Troubleshooting is still complicated:
  – Logs provided by the SDN stack is still poor
Architecture - Before

- Single FSFW interfacing all apps
- Troubleshooting done through logs and tcpdump captures
- A testing methodology in place before adding new testbeds:
  - Understanding of the researcher’s applications
  - Tests in lab prior adding to the production environment
  - AmLight and Researcher manage the SDN app together
    - Risky
    - Very time-consuming
    - A few reloads happened, hard to understand “why”
New Architecture – Proof of Concept

- Two Layers of Virtualization
  - Main/Production Layer
  - Experimentation Layer

- Experimentation Layer had a “Sanitizer” module added:
  - Controls what OpenFlow messages can be sent to the “Physical Layer”
  - Allows filters per OpenFlow Type, per-match and per-action
  - Off-loads switches from unsupported OpenFlow messages

- Sanitizer logs transactions and filters based on dictionaries:
  - XML files created as result of OF Tests
  - Detailed logs per slice or per type of message

- OpenFlow Sniffer keeps monitoring all communication
  - To help vendors in their troubleshooting activities
Methodology

• OF Tests:
  – Each device, software version and line card type is stressed in lab
  – Unsuccessful tests are collected and processed
  – When a specific match or action is not supported, it is added to the dictionary

• XML filters
  – Defines the Dictionary to be used by Sanitizer
  – They can be created through field experience

• Filters are stateless:
  – Less powerful but easier to deploy and faster
  – Some issues require stateful filters (future work?)
Examples

• ONOS vs Brocade CES
  – ONOS sends all flows in a single batch command
  – Brocade CES doesn’t support MAC rewrite
  – ONOS logs only have “batch failed”
  – Tcpdump had to be used
  – Satinizer’s dictionary has a “CES and Mac-rewrite don’t mix” entry and log it

• Brocade CES NI 5.7 vs OpenFlow Vendor type
  – Some OpenFlow messages type Vendor were forcing Brocade CES to restart the OpenFlow connection
  – Satinizer’s Dictionary has a “CES 5.7 doesn’t take unknown Vendor ID” filter and log it

• OESS Forwarding Verification vs Brocade MLX-4 4x10G line card
  – Ethertype 0x88bc not support, internal trace logs rotating too fast
  – Satinizer’s Dictionary has a “LP 4x10G and Etype “A” don’t mix” filter and log it
Findings

• Off-loading some filters help switches to focus on “supported” features
  – Also preserves switches internal trace logs queue

• New per-slice logging helps to identify which application sent a specific OpenFlow message
  – Helps researcher to improve his/her SDN application

• Troubleshooting logs helps vendors to reproduce the issue

• A testing methodology before adding anything to production is still required, once some issues require stateful/complex filters
Future

• Testbed Sanitizer was a proof-of-concept to understand how complex and deep the problem is

• Future is unclear: should we develop a production sanitizer? Or should we “force” vendors to create a better code?

• Stateful filters are very important, but they are very complex to deploy

• OF 1.3 will be even more complicated: meters, multi-tables, etc.