AmLight’s OpenFlow Sniffer dissected: Troubleshooting production networks

Jeronimo Bezerra, Julio Ibarra
Florida International University
{jbezerra,julio}@amlight.net

Humberto Galiza, Marcos Schwarz
Rede Nacional de Ensino e Pesquisa
{humberto.galiza,marcos.schwarz}@rnp.br
Outline

• Context
• Motivation
• Features
• Outputs
• Roadmap
AmLight is a Distributed Academic Exchange Point

- Production SDN Infrastructure (since Aug 2014)
- Connects AMPATH and SouthernLight GOLES - GLIF Open Lightpath Exchanges
- 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 enabled
    - Including AMPATH and SouthernLight
  - Currently 5 slices for experimentation (including ONOS SDN-IP)
Context (2)

Zabbix server: Openflow - Total Number of Flows - All Slices (7m 19d 23h)

- AmLight’s NRENs
- Other NRENs
- IDCP
- NSI
- OSCARS
- OpenNSA

Northbound: Users’ APIs
Motivation

• As troubleshooting SDN is still complex, a few tools are being developed at AmLight:
  – Testbed Sanitizer
  – **An OpenFlow Sniffer**
  – A multi-slice SDN Traceroute
  – Integration tools: Zabbix NMS w/ OESS and FSFW

• Why a new OpenFlow sniffer?
  – Wireshark requires X or capture/send and dissector for OF
    • OF 1.0: < 50% dissected
  – Tshark uses Wireshark dissectors
  – There are other tools, but they are not specific for real time and command line OpenFlow troubleshooting (lack of OpenFlow filters)
Features

- OpenFlow 1.0 support
- Completely passive/libpcap
- Runs on Linux shell
  - No need for X Windows
- Colors important user fields
- Easy to install *(install python-pcapy && git clone)*
- Supports OpenFlow type filtering using a JSON file
- Converts FlowMods to OVS-OFCTL commands
  - Help “reproduce” some problems
- Apache License
- https://github.com/jab1982/ofp_sniffer
Outputs (1/2)

OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 88 XID: 4959165
4959165 OpenFlow Match - wildcards: 4194300 dl_vlan: 1116 in_port: 4
4959165 OpenFlow Body - Cookie: 0x00 Command: Add(0) Idle/Hard Timeouts: 0/0 Priority: 32768 Buffer ID:
4959165 OpenFlow Action - Type: SetVLANID Length: 8 VLAN ID: 3221 Pad: 0
4959165 OpenFlow Action - Type: OUTPUT Length: 8 Port: 67 Max Length: 65535

2015-09-15 11:10:29.658553 10.0.2.15:44950 -> 190.103.187.35:6633 Size: 126
OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 72 XID: 2
2 OpenFlow Match - wildcards: 3678453 dl_vlan: 31 dl_dst: 10:00:00:01:20:00
2 OpenFlow Body - Cookie: 0x00 Command: Delete(3) Idle/Hard Timeouts: 0/0 Priority: 32768 Buffer ID:
ovs-ofctl del-flows tcp:190.103.187.35:6633 "priority=32768 dl_vlan=31,dl_dst=10:00:00:01:20:00"

2015-09-14 19:00:49.591812 190.103.187.35:6633 -> 10.0.2.15:44797 Size: 66
OpenFlow Version: 1.0(1) Type: Error(1) Length: 12 XID: 2
2 OpenFlow Error - Type: BadRequest Code: BadVendor

2015-09-15 11:10:29.736198 190.103.187.35:6633 -> 10.0.2.15:44950
OpenFlow Version: 1.0(1) Type: BarrierRes(19) Length: 8 XID: 3
3 OpenFlow Barrier Reply
2015-09-15 08:33:18.349577 190.103.187.35:6633 --> 10.0.2.15:44835 Size: 362
OpenFlow Version: 1.0(1) Type: StatsRes(17) Length: 308  XID: 4
4 StatRes Type: Flow(1)
4 StatRes Length: 96 Table_id: 0 Pad: 0
4 StatRes  OpenFlow Match - wildcards: 3678447 dl_type: 0x806
4 StatRes duration_sec: 372922, duration_nsec: 889000000, priority: 1, idle_timeout: 0, hard_timeout: 0
4 StatRes Length: 96 Table_id: 0 Pad: 0
4 StatRes  OpenFlow Match - wildcards: 3678447 dl_type: 0x88cc
4 StatRes duration_sec: 372922, duration_nsec: 889000000, priority: 100, idle_timeout: 0, hard_timeout: 0
4 StatRes Type: Flow(1)
4 StatRes Length: 104 Table_id: 0 Pad: 0
4 StatRes  OpenFlow Match - wildcards: 3678455 dl_dst: 10:00:00:01:20:00
4 StatRes duration_sec: 68231, duration_nsec: 128000000, priority: 32768, idle_timeout: 0,
Handling Network Virtualization (1/2)

- Supporting Network Testbeds is a new trend
  - But creates another layer
- Applications don’t talk to OF switches directly
  - Virtualization layer interfaces both entities
- Network Sniffers don’t see the end-to-end flow:
  - Or it sees OF switch talking to Virtualization Layer
  - Or it sees Virtualization Layer talking to Application
- OpenFlow messages don’t identify the OF switch:
  - How to associate OF switch to Application?
    - Specially for OFP_ERROR messages?
Handling Network Virtualization (2/2)

1. OpenFlow Message Received
   - Is a PacketOut with LLDP?
     - Yes
       - Collect source IP, source TCP port and the DPID
       - Add to the Association Dictionary
     - No
       - Parse OpenFlow Packet
         - Check if the source IP and TCP Port is in the Association Dictionary
           - Found it?
             - Yes
               - Print Packet with the OpenFlow device’s name
             - No
               - Print Packet without specifying the OpenFlow device
Handling Network Virtualization (2/2)

OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 72 XID: 21
Match - wildcards: 0x2fffeff dl_type: ARP(0x806)
Body - Cookie: 0x100005d483efc Command: Delete(3) Idle/Hard Timeouts: 0/0
Body - Priority: 32768 Buffer ID: 0xffffffff Out Port: None(0xFFFF) Flags: SendFlowRem(1)

OpenFlow Version: 1.0(1) Type: BarrierReq(18) Length: 8 XID: 21

OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 72 XID: 24
Match - wildcards: 0x2fffeff dl_type: ARP(0x806)
Body - Cookie: 0x10000c58802b5 Command: Delete(3) Idle/Hard Timeouts: 0/0
Body - Priority: 32768 Buffer ID: 0xffffffff Out Port: None(0xFFFF) Flags: SendFlowRem(1)
Roadmap

- **Version 0.3 – By June 2016**
  - Full OF 1.3 (.5) support
  - Read from Libpcap files
  - Better documentation
  - Better code organization
  - Support for virtualization
  - Interface for extra filters

- **Version 0.4 - ?**
  - Full NICIRA/OVS support
  - SSL/TLS support
  - Traffic Profile?
  - Suggestions??
Use Cases

• Teaching/Learning:
  – Great tool to teach/learn SDN and OpenFlow
  – Easy to see all OpenFlow messages and fields

• Coding:
  – Great way to see if your controller (Ryu, POX, ONOS) is sending the OpenFlow message the way you expect
  – Example: Malformed OF messages are not send by Ryu and no alarm is generated

• and Troubleshooting:
  – SDN networks are very hard to debug: lack of tools, protocols and logs
  – Most OF switch agents are in a beta deployment phase

• More information:
  – www.sdn.amlight.net
  – Papers, Presentations, Videos, etc.
AmLight’s OpenFlow Sniffer dissected: Troubleshooting production networks

Questions?

Jeronimo Bezerra, Julio Ibarra
Florida International University
{jbezerra,julio}@amlight.net

Humberto Galiza, Marcos Schwarz
Rede Nacional de Ensino e Pesquisa
{humberto.galiza,marcos.schwarz}@rnp.br