AmLight EXP
Americas Lightpaths Express & Protect

TNC17– Linz, Austria – May 31st 2017

Handling Network Events in a Production SDN Environment

Jeronimo Bezerra <jbezerra@fiu.edu>
Florida International University

Outline

- Introduction to AmLight
- SDN Topologies
- Troubleshooting production SDN networks
- What should be monitored?
  - Control Plane Monitoring
  - Data Plane Monitoring
- Tools and Approaches used @ AmLight
- Future

AmLight: a Distributed Academic Exchange Point

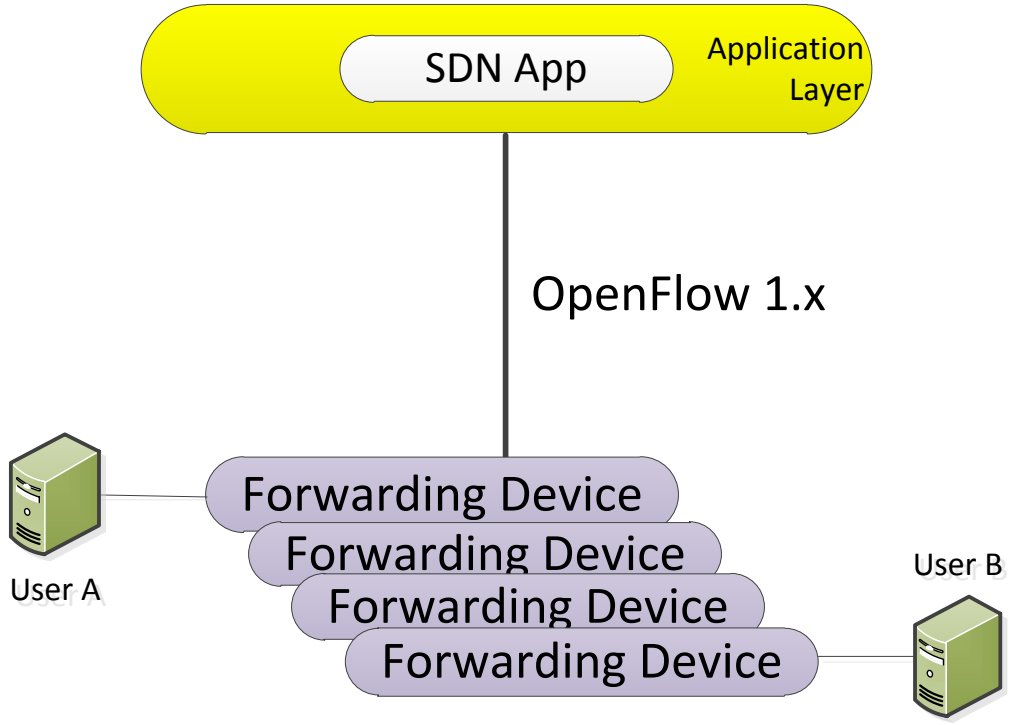
- Production SDN Infrastructure since Aug-2014
- Collaboration: FIU, NSF, ANSP, RNP, Clara, REUNA and AURA
- Connects North and South America with multiple 10G and 100G links
  - 4 x NAPs: Brazil(2), Chile and Panama
  - 2000+ institutions connected
- Carries Academic and Commercial traffic
- Control Plane: OpenFlow 1.0
- Network Programmability/Slicing
  - OESS/NOX, ONOS, Kytos and Ryu
- NSI-enabled
- Currently, operating with more than a 1000 flow entries
- Web site: www.sdn.amlight.net

Troubleshooting a production SDN network

- Troubleshooting production environments has different requirements
  - Has to be agile, least disruptive as possible and needs historical data
  - Tools have to be handy
- With SDN, legacy troubleshooting tools are partially useful or completely useless
  - OAM (Operation, Administration and Maintenance) is not supported by OpenFlow (yet)
  - Ping, traceroute, SNMP, Wireshark/Tcpdump are not made for OpenFlow
- More than ever, deep knowledge of the hardware and software platforms are required:
  - Usage of the ”hidden” commands and application logs become part of your routine
- A ”premium” support contract with hardware vendor is desired
  - Going through the level 2 TAC team will increase your stress and the network recovery time

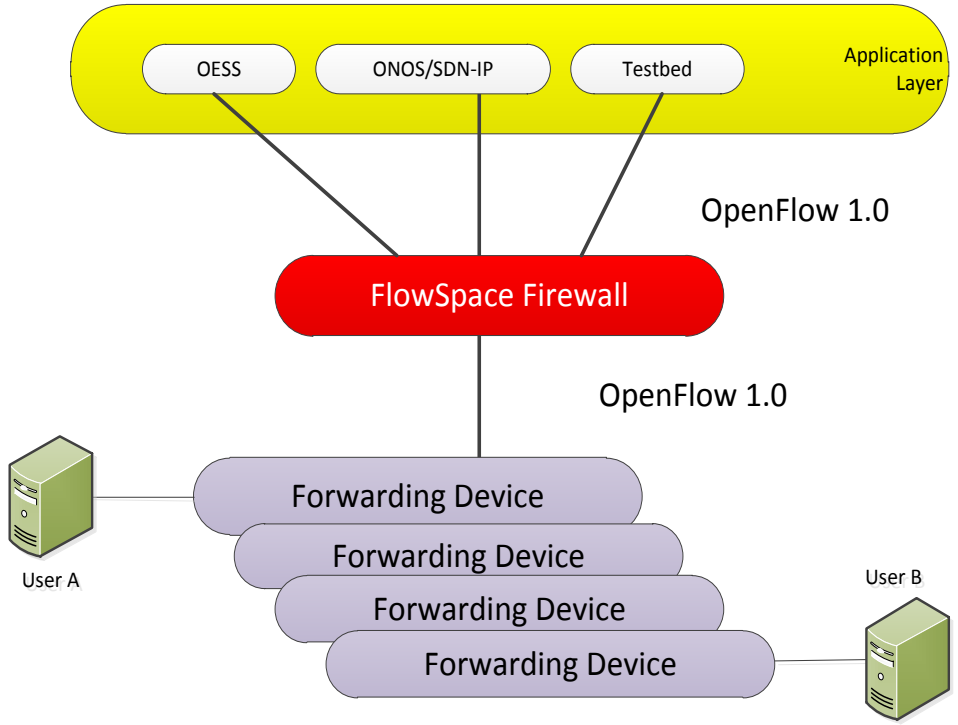
SDN Topologies: Starting Simple

- Usually, with just one SDN App, troubleshooting is less complex
  - One SDN App is connected through an out-of-band network to multiple OF switches
  - SDN App has full control of ports and VLANs
- A good network sniffer and a centralized Syslog server are the key to success here
  - Helps validate the OpenFlow messages sent and received
  - Easy access to event messages



SDN Topologies: Adding Complexity

- When supporting control planes in parallel you have:
  - More applications to understand and track
  - Different levels of software stability
  - Higher chances of network outages
- Slicing/Partitioning adds complexity:
  - OpenFlow communication between OpenFlow switch and SDN App is not end-to-end:
    - OF Switch -> Slicer + Slicer -> OF App
  - Complexity to track which switch is talking to which SDN App and vice-versa
    - OFPT_ERROR messages are asymmetric
    - OF doesn't carry DPID on each OF message
- "Traditional" sniffers are not enough to track *indirect* OpenFlow messages



Control Plane: What should be monitored?

- Everything concerning the OpenFlow communication:
  - # of flows installed
    - Avoid getting close to the limits documented (weird stuff might happen)
  - Rate of FlowMods, PacketOut/PacketIn and Stats Requests / second:
    - Switch's CPU is directly affected by these rates
  - # of OFP_FLOW_ERROR messages:
    - Some messages might indicate that a crash is about to happen (FULL_TABLE)
  - Flows duration:
    - Helps to understand traffic disruption due to flows being reinstalled
  - Flow and Port Counters (bps and pps)
- If slicing/virtualization is a reality, collect counters per slice
- Most of the SDN apps don't provide such data, some provide through REST interfaces

Data Plane: What should be monitored?

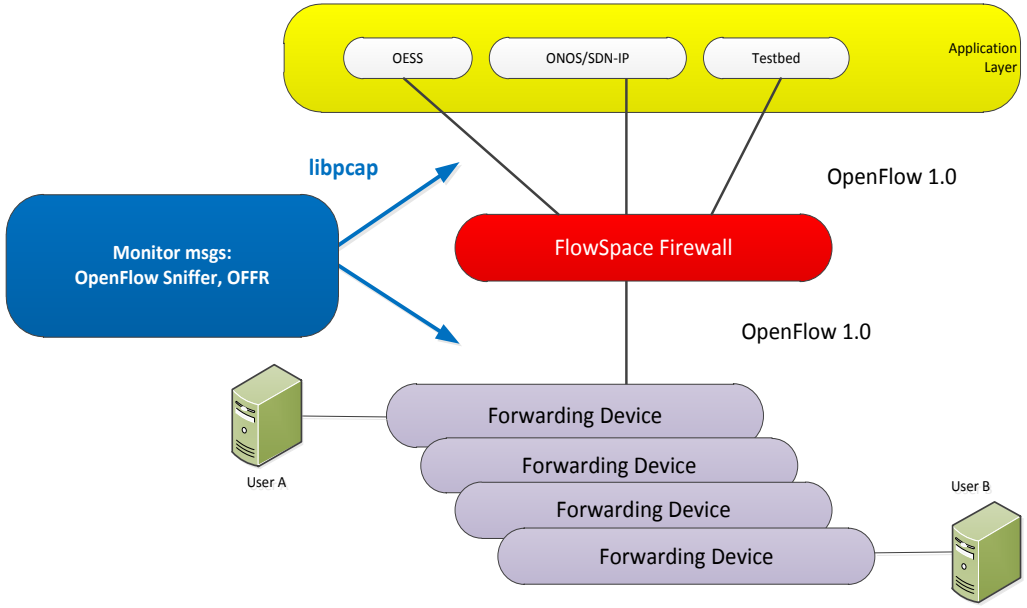
- In some cases, OpenFlow rules are installed but traffic is not flowing: *black holes*

- Some possible data plane black holes:
  - A specific line card or interface discarding all traffic
    - Due to an interface memory issue, flows are installed but traffic is discarded
  - Interface down in one side but up in the remote and the SDN App doesn't understand that
    - For instance: 10G LAN-PHY, Ethernet circuits and 100G long haul circuits
    - In this case, depending of the side, the SDN App installs the circuits pointing to the affected link, discarding all traffic
  - A specific installed flow entry crashed
    - Due to an interface memory issue, one specific flow is affected and traffic is discarded
    - Depending of the number of OpenFlow switches and flow entries, finding the problem might be extremely time-consuming

- In these cases, in-band tests are required:
  - Just a very few SDN Apps test in-band per link
  - No SDN Apps test in-band per flow

Control Plane Monitoring: Tools

- Monitoring the OpenFlow messages with passive packet capture:
  - Non-intrusive/Almost risk-free
- Few tools available:
  - Wireshark/tshark/tcpdump
  - AmLight OpenFlow Sniffer
- AmLight OpenFlow Sniffer was created to be CLI-based with support to environments with *slicers*:
  - Dissects OpenFlow 1.0 and 1.3*
  - Doesn't require GUI or XWindow
  - End-to-end communication visualization
  - Highlights important fields
  - Many filters available to optimize tshoot!
  - Source: *github.com/amlight/ofp_sniffer*

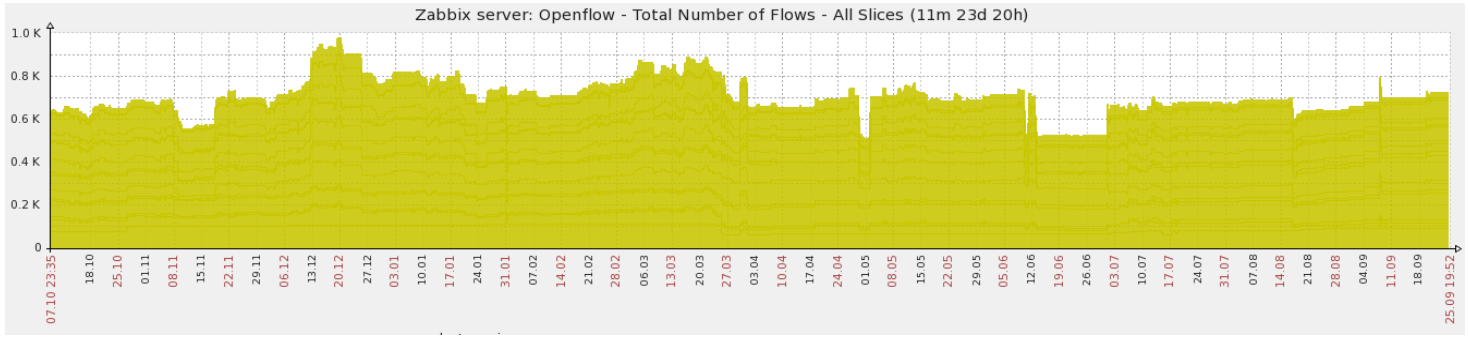


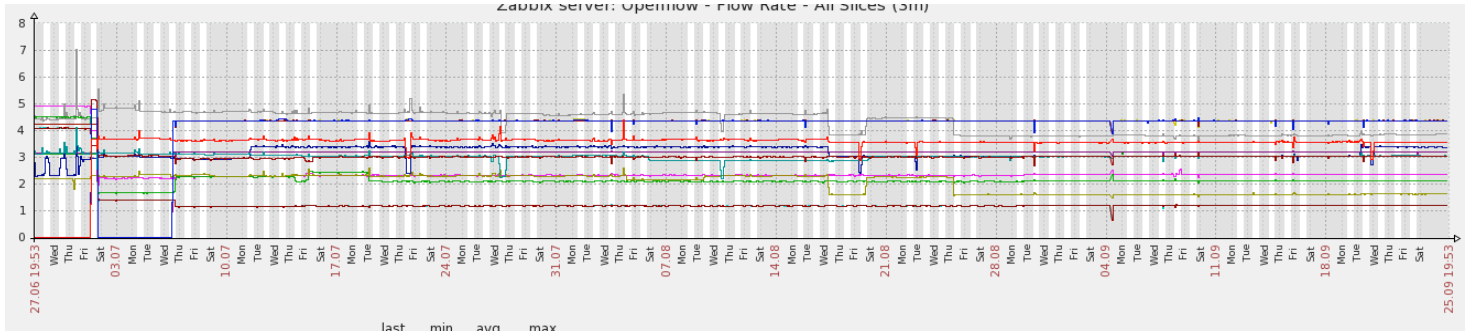

```
2016-01-22 16:42:52 OF_Controller:6633 -> OF_Switch:32975 Size: 146 Bytes
OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 80  XID: 20
OpenFlow Match - wildcards: 3276782 dl_type: 0x800 nw_dst: 10.10.11.0/25 in_port: 53
OpenFlow Body - Cookie: 0x00 Command: Add(0) Idle/Hard Timeouts: 0/0
        Priority: 32768 Buffer ID: 0xffffffff  Out Port: 65535 Flags: SendFlowRem(1)
OpenFlow Action - Type: OUTPUT Length: 8 Port: 8 Max Length: 0
```

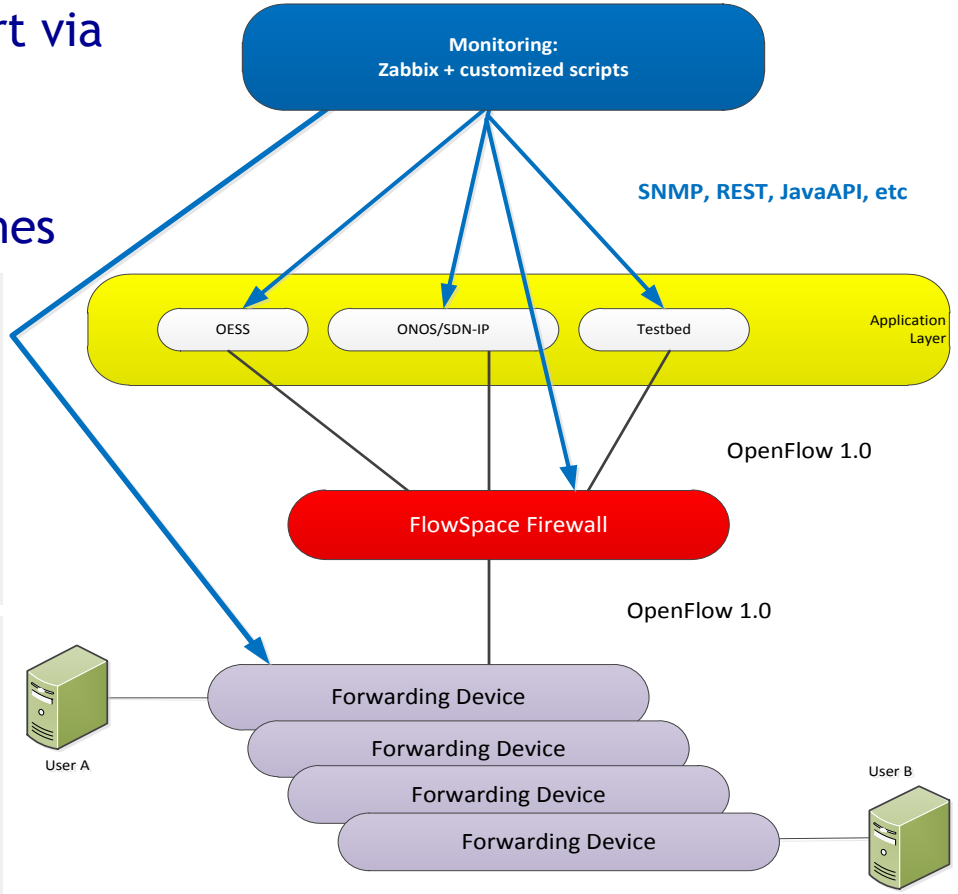
Control Plane Monitoring: Tools [2]

Monitoring All Applications and Counters in a centralized NMS:

- Scripts collect info from SDN Apps' REST interfaces and export via JSON
- Zabbix imports JSON data and save into a MySQL database
- Currently, collecting data from OESS, ONOS, FSFW and switches

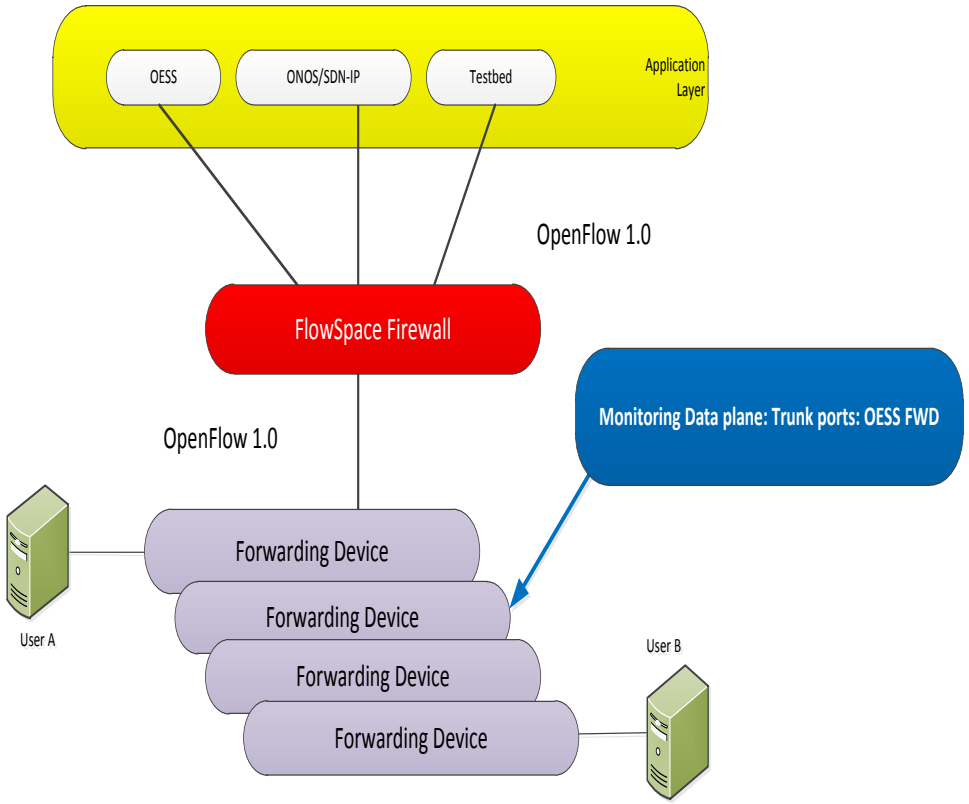






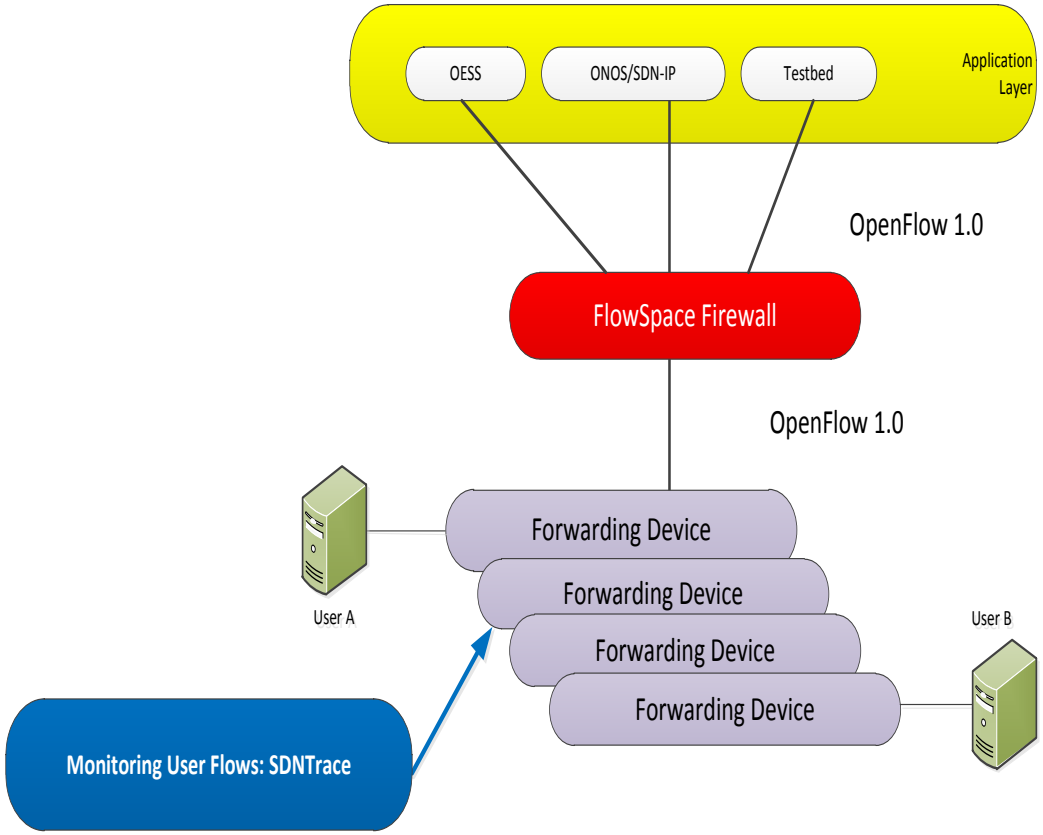
Data Plane Monitoring: Tools

- Most of the SDN Apps use LLDP or BDDP for topology discovery
  - Once the topology is discovered, these protocols are not used to monitor the topology
  - Also, interval between LLDP/BDDP packets is not appropriated for link monitoring
- An in-band testing approach is needed to validate the Data Plane
  - OESS does through its Forwarding Verification module
  - Most of other SDN Apps don't have anything equivalent
- Even though OESS/FVD validates the data path, it doesn't valite users' flows
  - A full port issue is detected, but a single flow issue is not



Data Plane Monitoring: Tools [2]

- Monitoring individual flows is important but extremely complex
  - Being proactive with all flows is desired but the interval between tests and number of flows needed must to be taken into consideration
  - Using a mix approach is the best suggestion
    - Track ”most important” flows only
    - Users won’t be happy, but your switches won’t crash
- An approach to test users’ flows was developed at AmLight (next)



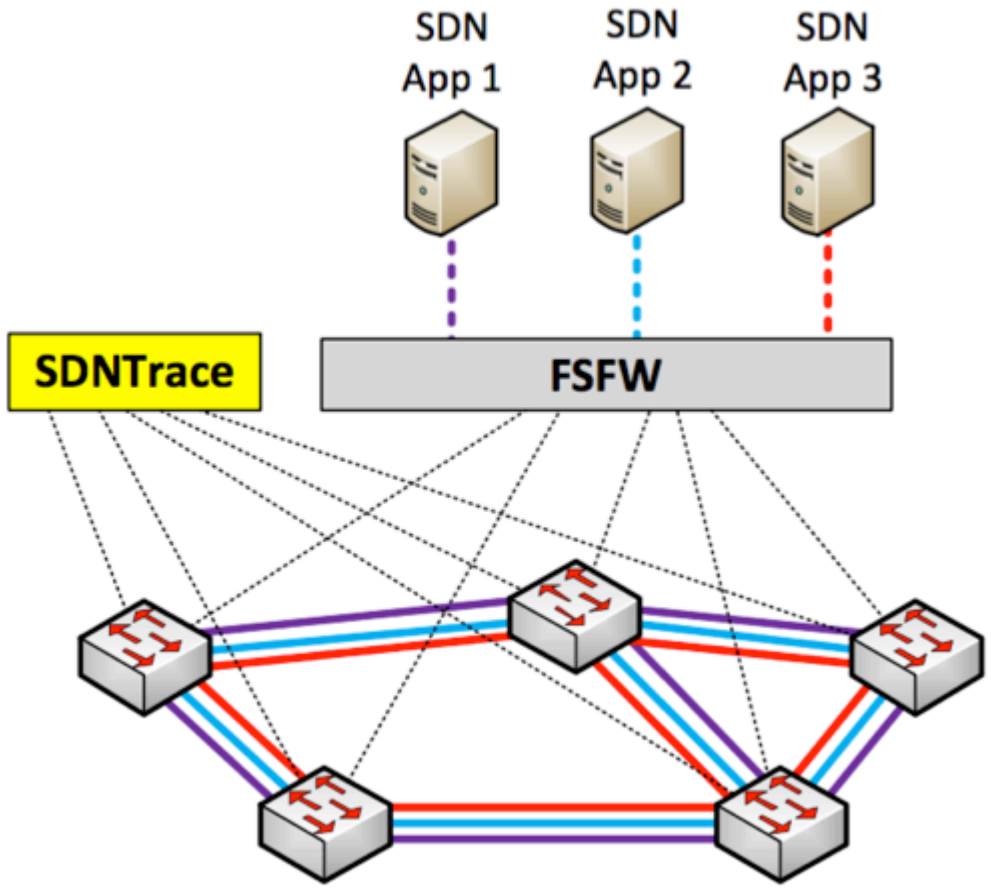
AmLight ExP
Americas Lightpaths Express & Protect

Data Plane Monitoring: Tools [3]

- AmLight's developed its own SDNTrace to test users' flows without changing them
  - Works through GUI or REST
  - Very lightweight
  - Very "cheap", only two-four flow entries needed
  - Traces L2 and L3 flows
  - Developed in collaboration with the **Academic Network of Sao Paulo/Brazil**
  - Supports INTER-DOMAIN tracing!

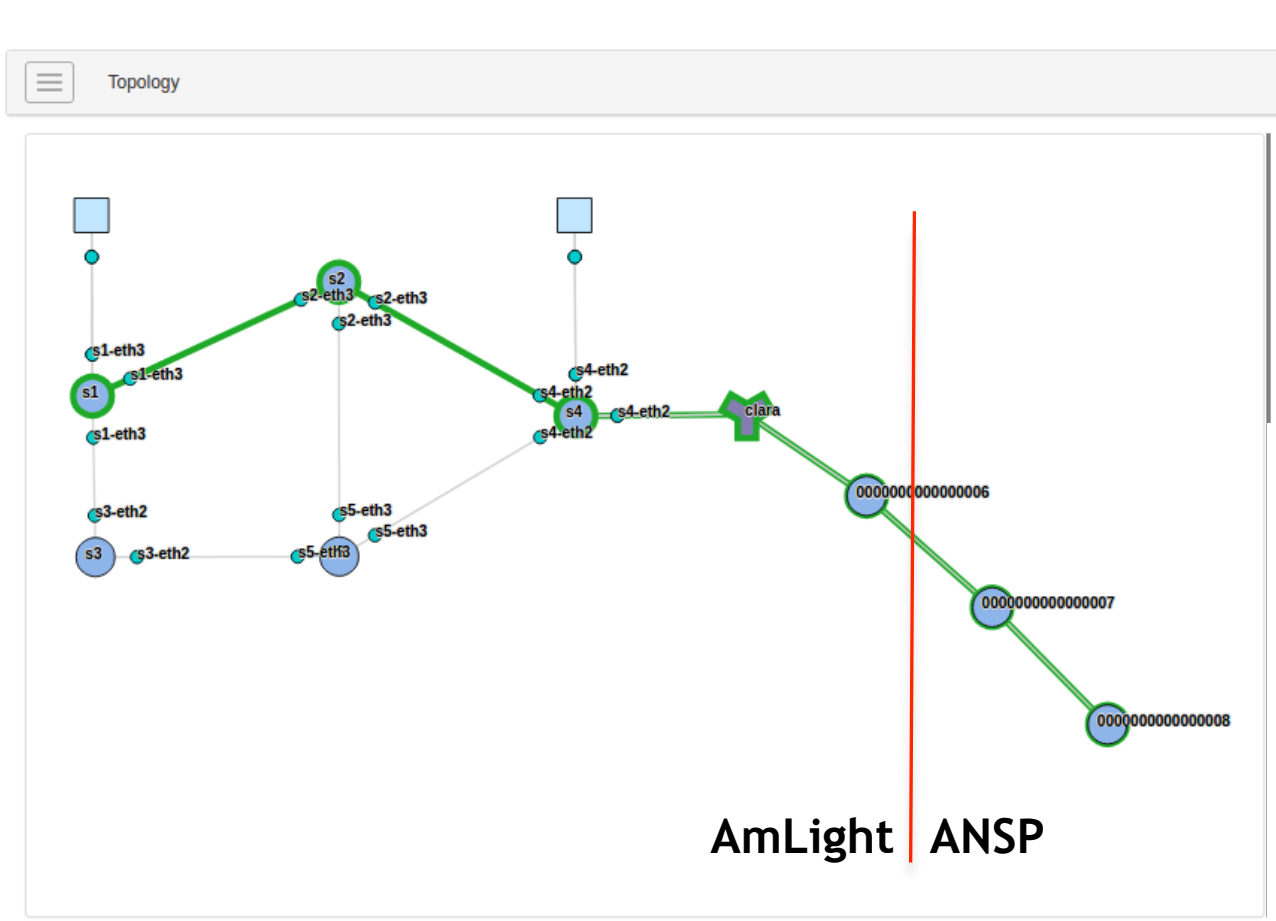
- Tracing a circuit is done in seconds instead of many minutes and can be easily integrated with Zabbix or Nagios

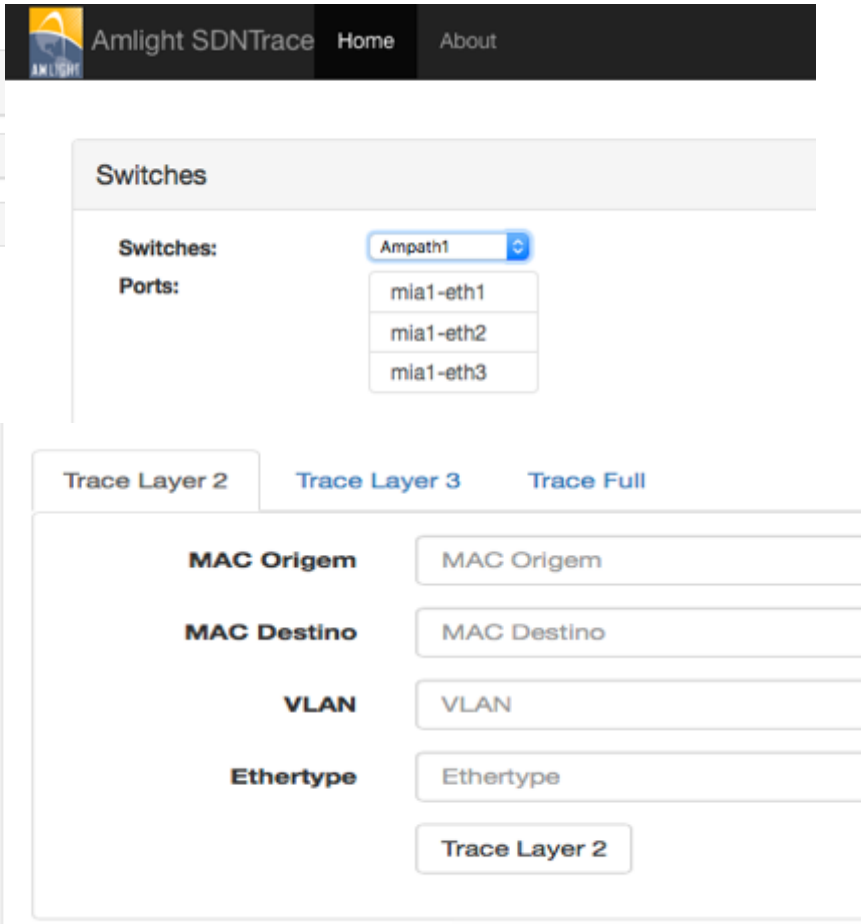
Available at: github.com/amlight/SDNTrace



AmLight ExP
Americas Lightpaths Express & Protect

Data Plane Monitoring: Tools [4]





AmLight ExP
Americas Lightpaths Express & Protect

Future: SDN Looking Glass

- Central point for SDN troubleshooting
- It will centralize all monitoring and troubleshooting information being slice/app-independent and:
  - Store all statistical data (flow, ports, etc.) and OpenFlow messages into a persistent repository (SQL)
  - Track real time OpenFlow Control Plane messages using the AmLight's OpenFlow Sniffer
  - Track non-OpenFlow information (CPU/Memory utilization, for instance) using SNMP/SSH
  - Run data plane traces, including inter-domain traces, automatically
  - Generate alerts in case of Data Plane black holes
  - Take network snapshots: save the network state for future troubleshooting and capacity planning
  - Provide REST to be used by external SDN apps, auditing tools and external NMZ
- Collaboration with State University of Sao Paulo / Kytos SDN framework developers:
  - Kytos SDN framework was build with troubleshooting in mind, helping the SDN operation
- Launch date: Internet2 Technology Exchange 2017 (October 2017)

THANK YOU!

Jeronimo Bezerra
jbezerra@fiu.edu

Handling Network Events in a Production SDN Environment
TNC2017