Software-Defined Networking @ AmLight

Jeronimo Bezerra
<jbezerra@fiu.edu>

10/12/2016

Outline

- Who we are
- What is SDN and OpenFlow?
- Justification and Experience
- Lessons Learned
- Conclusion

Who we are

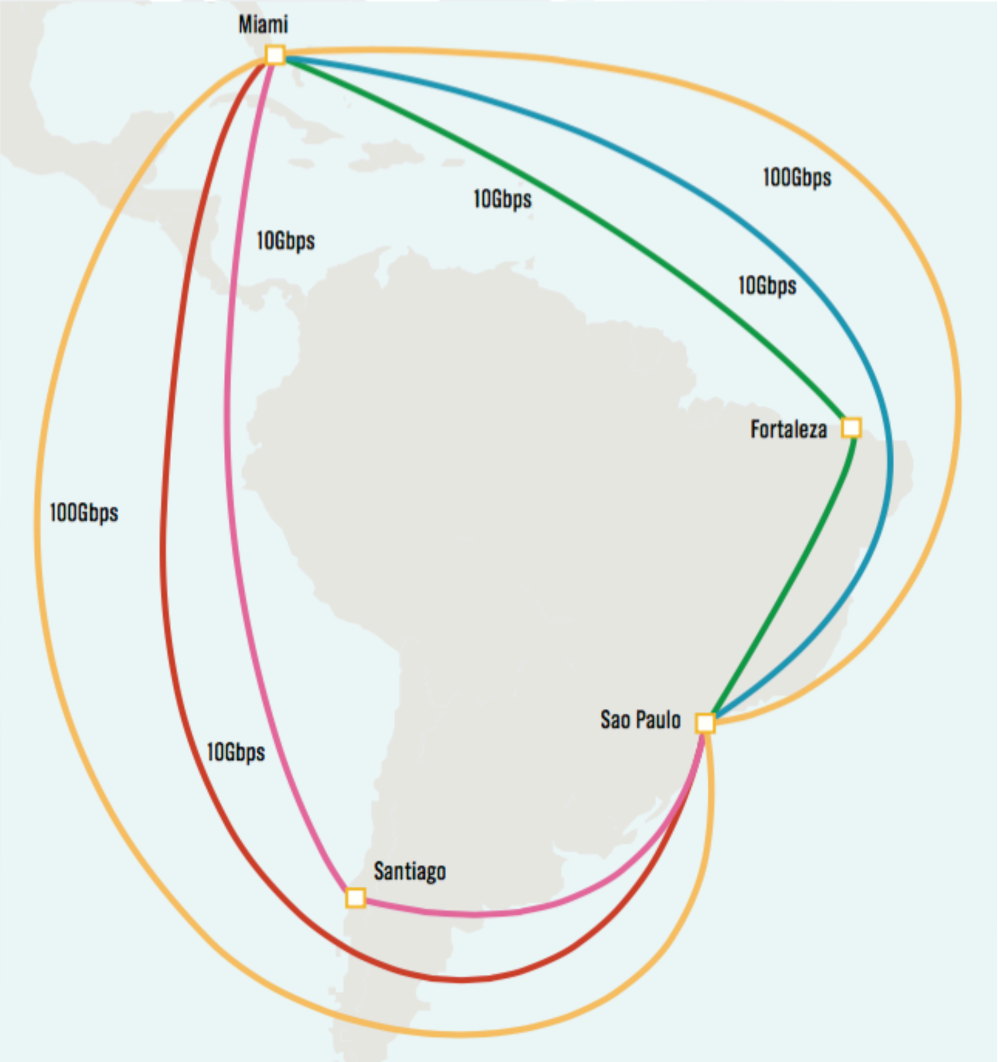
- CIARA:
  - Center for Internet Augmented Research & Assessment
  - CIARA assesses and measures FIU's effectiveness in the use of technology to augment the rate of discovery for domain researchers

- AMPATH(Miami) and SouthernLight(Brazil):
  - Academic Internet Exchange Points

- AmLight:
  - Network that connects AMPATH to SouthernLight

AmLight Today

- Four points of presence:
  - Miami
  - Sao Paulo/Brazil
  - Fortaleza/Brazil
  - Santiago/Chile
- Link configuration:
  - Two 100G links between Miami and Sao Paulo
  - 10G ring involving Miami, Fortaleza, Sao Paulo, Santiago and Miami
- Two topologies:
  - SDN ring (old Layer 2 ring)
  - MPLS ring



How did everything start?

Moving Towards SDN @ AmLight

4th Annual Global LambdaGrid Workshop

Oct 1st 2014

Queenstown, New Zealand

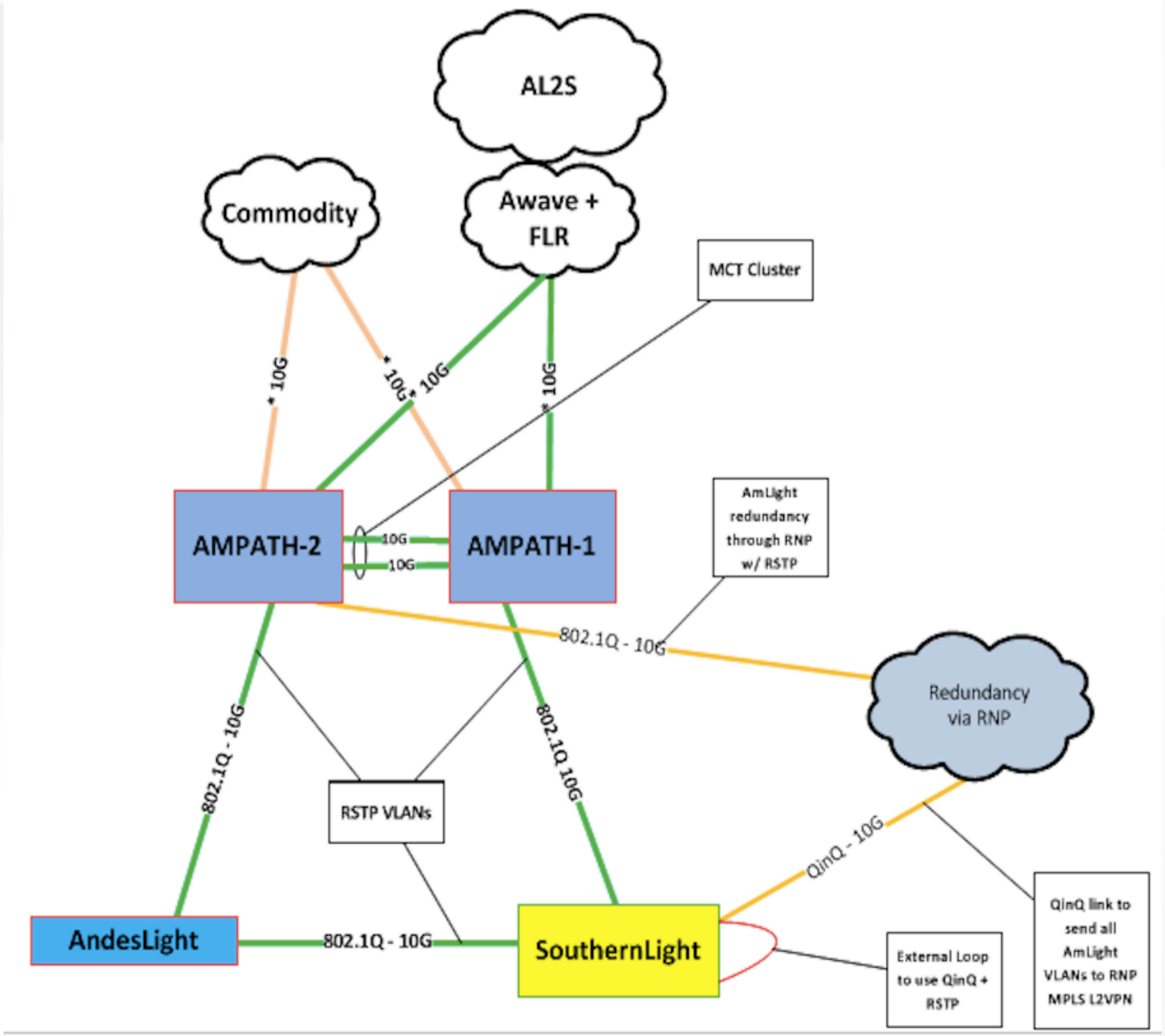
Jeronimo Bezerra <jbezerra@fiu.edu>

Back in time: AmLight in 2013

Layer 2 Ring detailed:

- A set of static VLANs
- Multiple instances of Brocade per-VLAN Rapid Spanning Tree Protocol (RSTP)
  - VLANs manually configured to take the shortest path to destination
- Provides redundancy to the MPLS ring with VLANs
  - Receives redundancy using QinQ
- Very resilient, 100% availability in 2013

AmLight in 2013 – Overview Diagram



It was working fine but…

AmLight Today– Drawbacks(1/2)

Layer 2 ring detailed again with its drawbacks:

- A set of static VLANs
  - Each new VLAN ID requires coordination between *N* parties, even from outside of AmLight
  - Each VLAN ID must be configured in all switches
  - Sometimes takes weeks to have a single VLAN working end-to-end
- Multiple instances of per-VLAN Rapid Spanning Tree
  - Brocade only supports 128 instances - 128 protected VLANs (34 in use today)
  - Each instance must be configured in all RSTP switches
    - Wrong configurations leads to loops
- Receives redundancy from the MPLS ring
  - Multiple Dedicated 10G port to accomplish this redundancy
    - Due to the characteristics of the configuration (L2VPN + QinQ)
    - High CAPEX

AmLight Today – Drawbacks(2/2)

- Very resilient, 100% availability in 2013
  - Semi-complex deployment and OPEX
  - Lots of meetings, hours, calls, emails exchanged to achieve the current configuration
- Other issues:
  - Users/Apps have no visibility of the network and its status
  - Every new protocol to be deployed leads to lots of meetings, maintenance windows, software upgrades, etc.

Let's talk about the future! So, how to …

1. **Improve operations efficiency?**
   - *How to handoff to the user the layer 2 provisioning inside AmLight?*
   - *How to improve layer 2 multi-domain provisioning?*

2. **Introduce network programmability?**
   - *How to support network testbeds managing the network infrastructure in a secure way?*
   - *How to add and evaluate new control planes in parallel?*

“Could SDN be a solution?”

Glossary

- *Switch/Bridge*
  - *Layer 2 or Link Layer device*
  - *Network device that forwards frames based on destination MAC addresses*

- *Router*
  - *Layer 3 or Network Layer device*
  - *Network device that forwards packets based on destination IP addresses*

- *Control Plane*
  - Plane responsible for choosing the best forwarding path (for frames or packets)
    - Many protocols available: Spanning Tree, QinQ, BGP, OSPF, RIP
  - For routers, a RIB (*Routing Information Base*) is created with the best routes
  - Works in the share CPU

- *Data Plane*
  - Plane responsible to forward frames between interfaces
  - Entries are installed in the FIB (Forwarding Information Base)
  - Dedicated and high performance CPUs are used: ASICs, FPGAs or CPU (small devices)

- *Management Plane*
  - Plane responsible for track devices' health
  - Export data using specific protocols SNMP, sFlow, NetFlow, IPFIX, …
  - Works in the share CPU

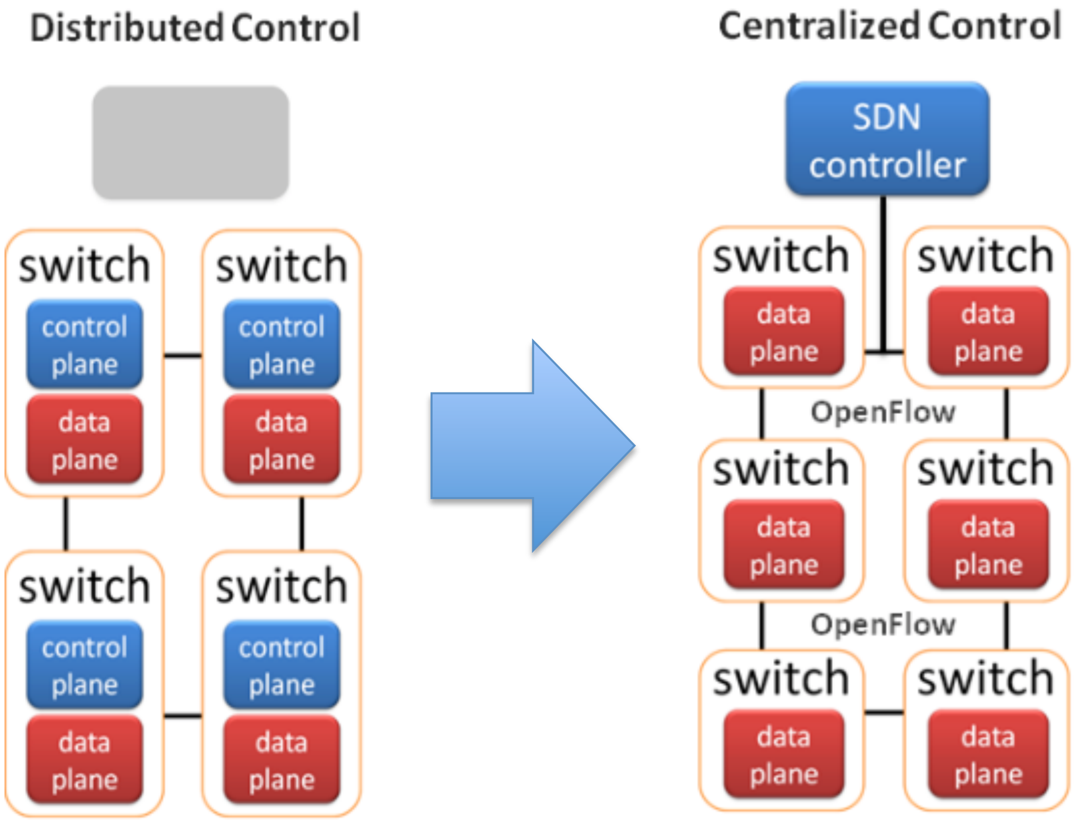
What is SDN? (1/3)

- *Software-Defined Networking* decouples Control Plane from Data Plane
  - Forwarding decision managed by an external controller
  - A standard protocol to manage the communication between the external network controller and the switches
    - OpenFlow 1.0 is deployed and supported by lots of vendors
    - Some vendors with support for *Hybrid Mode*
      - Some ports using OpenFlow, some ports using legacy protocol
    - A few switches with support for *Hybrid Ports*
      - OpenFlow and legacy traffic in the same port
      - Useful for an easy and incremental deployment

- It is in the beginning, but it has an interesting future

What is SDN? (2/3)

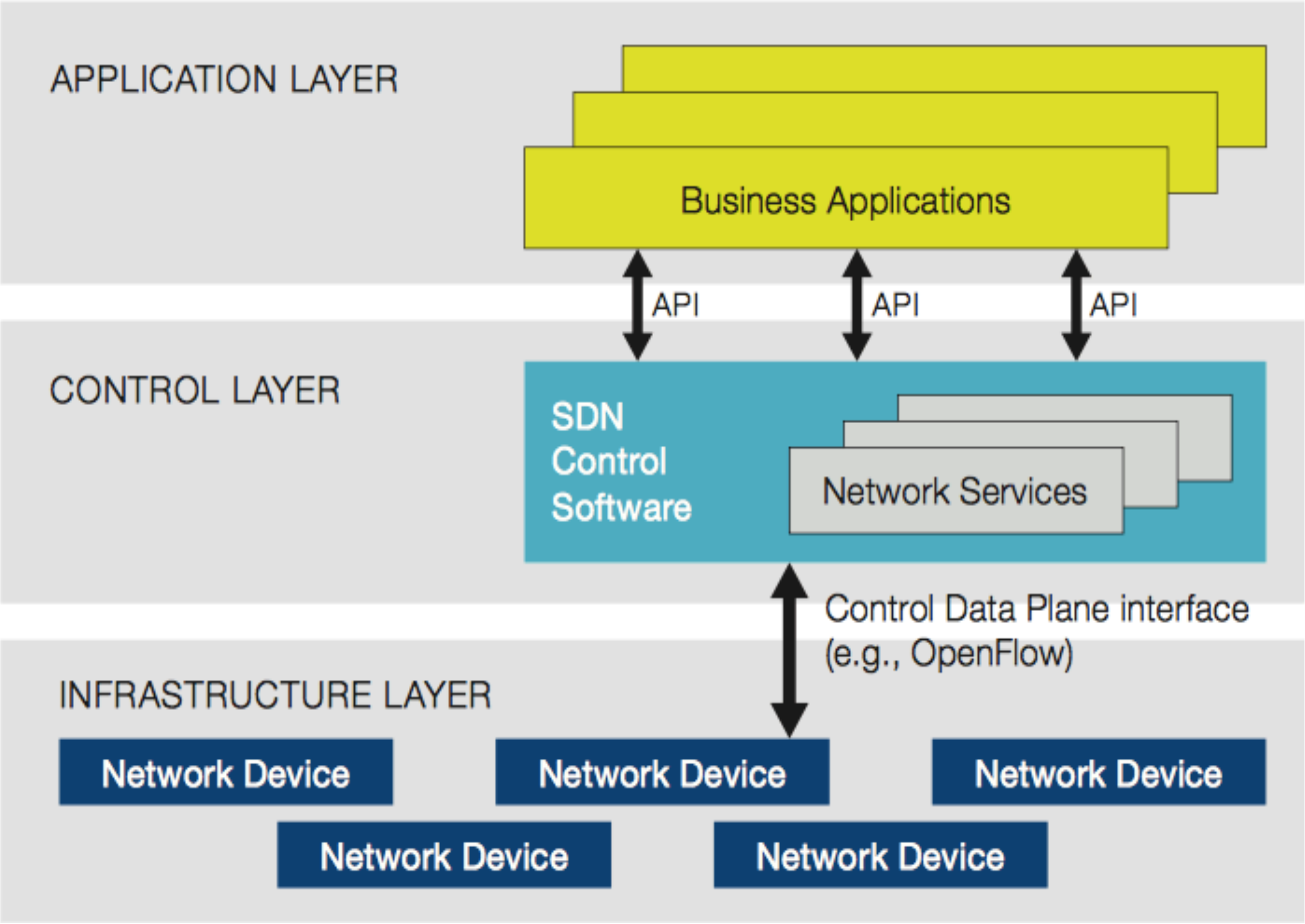
Without SDN:
STP, RSTP, MSTP, PBB, LDP, OSPF, IS-IS, MPLS, RSVP, RSVP-TE, iBGP, eBGP, PIM, MSDP might be needed in each device to decide the best path



With SDN:
Controller sees the full topology and chooses the best path

Figures from http://www.themetisfiles.com/2012/10/the-future-of-the-network-is-software-defined/

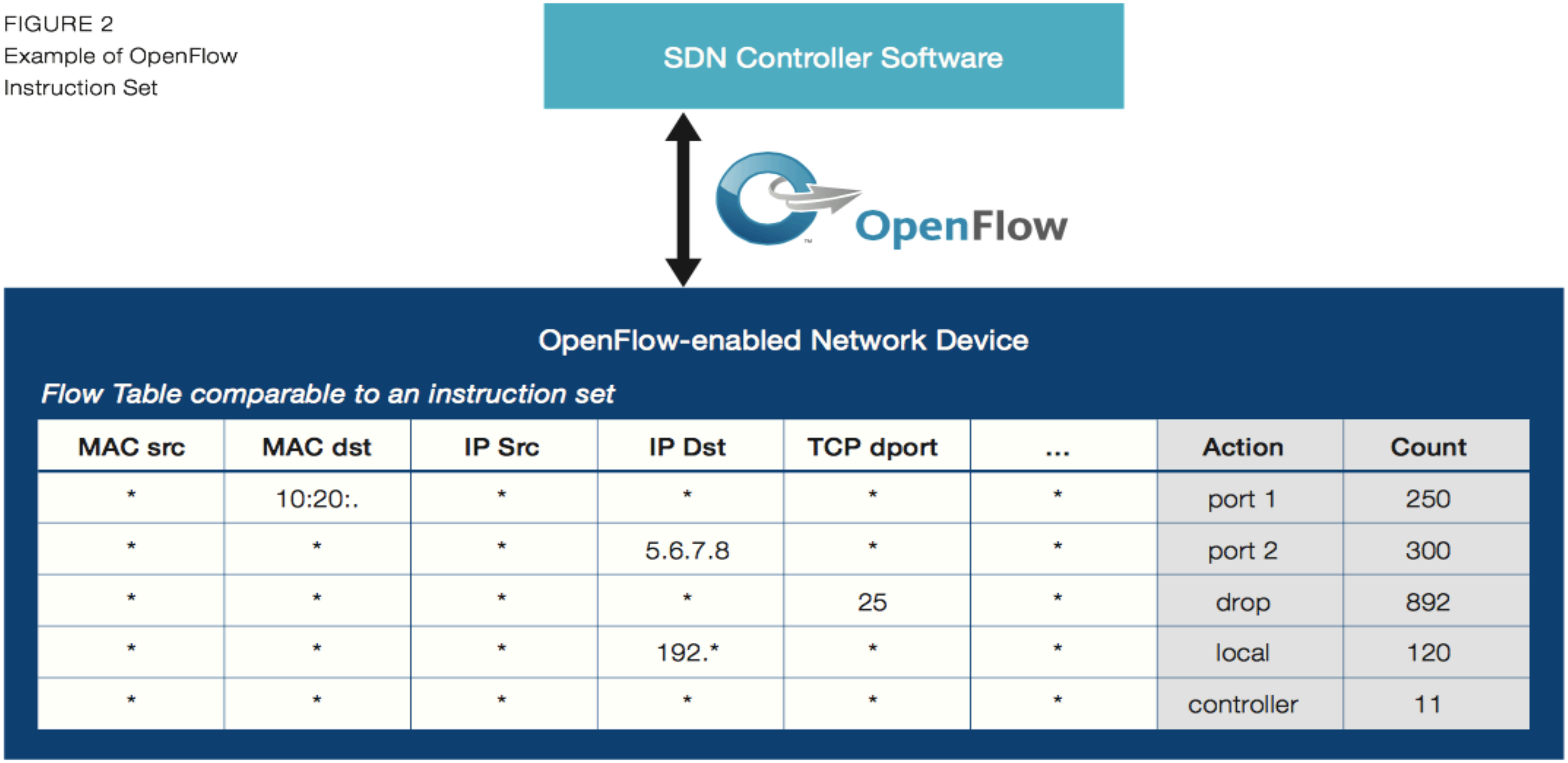
What is SDN? (3/3)



What is OpenFlow?

- OpenFlow is just an interface protocol (Southbound)
  - Used for controller to change the forwarding information table
  - It doesn’t not encapsulate traffic or create overheads
  - Currently with five versions: 1.0, 1.1, 1.2, 1.3, 1.4 and 1.5

FIGURE 2
Example of OpenFlow
Instruction Set


SDN Controller Software
OpenFlow
OpenFlow-enabled Network Device
Flow Table comparable to an instruction set

MAC src	MAC dst	IP Src	IP Dst	TCP dport	...	Action	Count
*	10:20:.	*	*	*	*	port 1	250
*	*	*	5.6.7.8	*	*	port 2	300
*	*	*	*	25	*	drop	892
*	*	*	192.*	*	*	local	120
*	*	*	*	*	*	controller	11

So, now that we “know” what SDN is, could SDN help AmLight?

Is SDN a possibility for AmLight?

- With SDN@AmLight, the network controller would be responsible for all network configurations (provisioning):
  - Network connectivity, including a loop-free topology
  - Rate-limits, priorization, statistics
  - And new services/deployments:
    - Security, new protocols, new applications, etc.

- Is it the right moment?
  - Openflow is a new protocol (4 years for OF 1.0 and <1 year for OF 1.4) but some important networks/players are deploying it:
    - Google, Facebook, Microsoft, Internet2, etc.
  - The academic community is extensively testing it
  - Vendors are providing very specialized support to all deployments
  - But, it's a new approach, new software code, require new skills, etc., so, it has risks
    - AmLight provides academic and commodity traffic to its users!
    - Almost all Openflow controllers are experimental

How to move toward SDN at AmLight? (1/2)

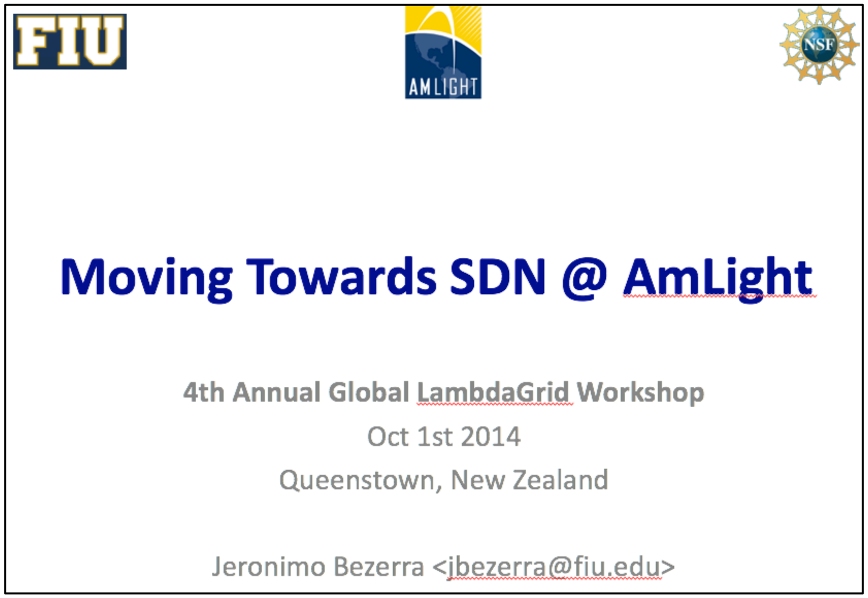
- The idea was to deploy SDN in a controlled environment, with some specific VLANs to start

- But, troubleshooting procedures would require new approaches and techniques, especially in a multi-domain environment
  - Open area

- It's a pre-requirement not losing some important network features we have today:
  - Flow statistics through sFlow
  - Low convergence time (sub-sec with per-VLAN RSTP)
  - Redundancy approaches

Ok, great! But how do we start this process?

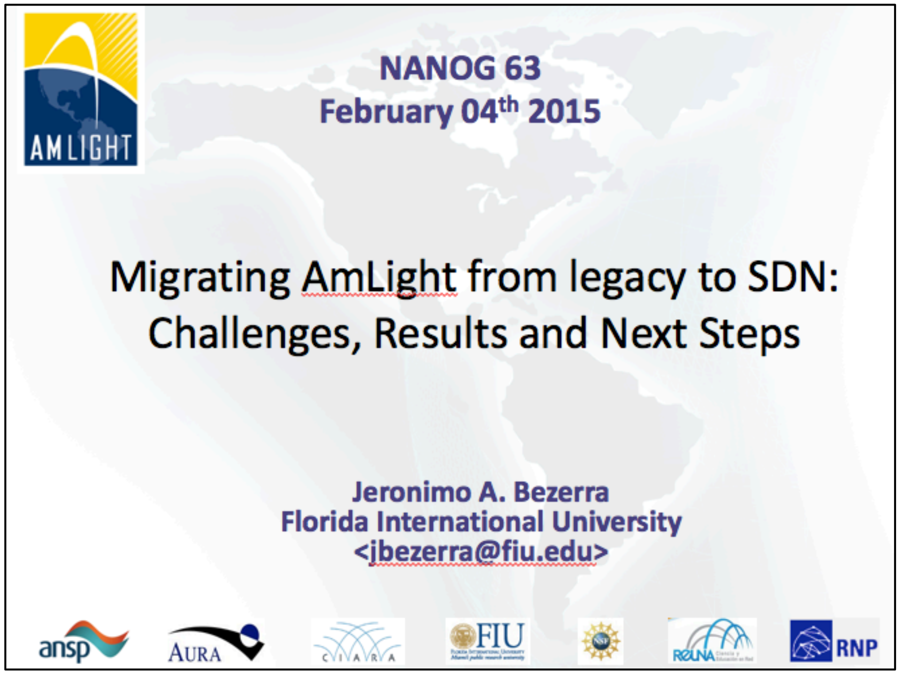
Step by Step to migrate to SDN:
AmLight Use Case



Moving Towards SDN @ AmLight

4th Annual Global LambdaGrid Workshop
Oct 1st 2014
Queenstown, New Zealand

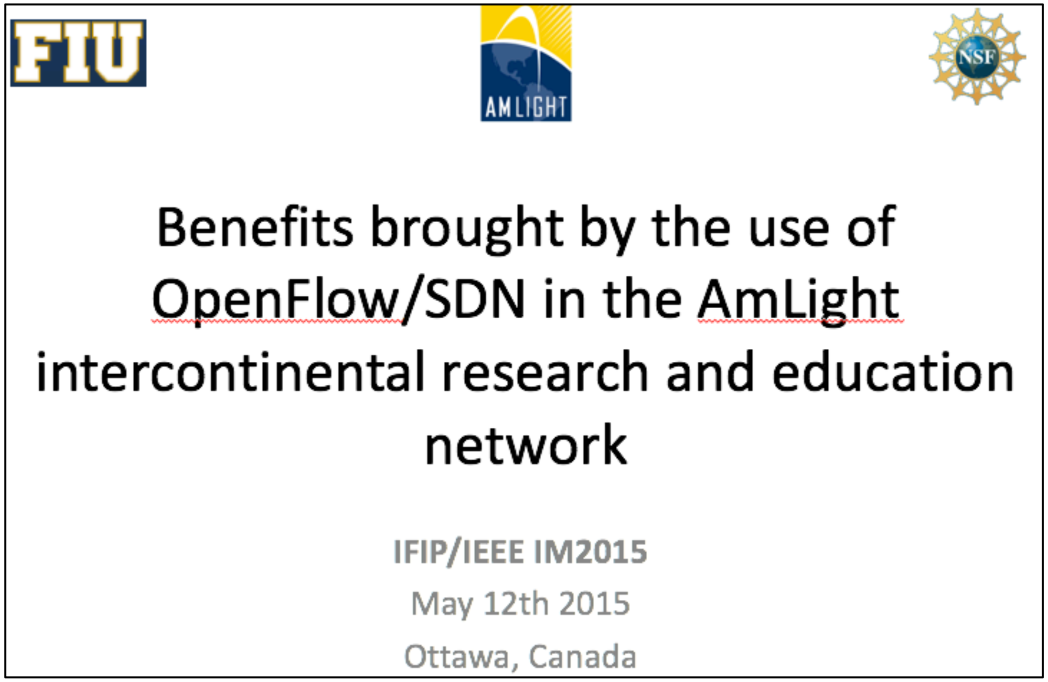
Jeronimo Bezerra <jbezerra@fiu.edu>



NANOG 63
February 04th 2015

Migrating AmLight from legacy to SDN: Challenges, Results and Next Steps

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



Benefits brought by the use of OpenFlow/SDN in the AmLight intercontinental research and education network

IFIP/IEEE IM2015
May 12th 2015
Ottawa, Canada

Step 1: Know your network

- Document everything you have in operation:
  - Link Aggregation, VLANs, MPLS, QoS, Port mirroring…

- Knowing what you have will help you choose controllers and applications
  - It doesn't mean you will find SDN applications that support everything!

- At AmLight we had:
    - Link Aggregation (Brocade Multi Chassis Trunk + LACP)
    - VLANs + Brocade per-VLAN Rapid Spanning-Tree
    - Port Mirroring

Step 2: Assessments of your devices (1/2)

- Supported protocols:
  - Openflow
    - Which version? 1.0, 1.1, 1.3 or 1.4?
  - Netconf, Yang?
- Openflow implementation phase:
  - Beta, Testing or stable?
- Openflow's *Optional* features
  - Metering, Port Group, LACP, etc.
- Is Hybrid port supported?

Step 2: Assessments of your devices (2/2)

AmLight use case:

- Brocade MLXe/XMR/CES switches:
  - Openflow 1.0 and 1.3 (1.3 started on Dec 2014)
  - Number of flows supported
    - MLXe (-D)/XMR: 64k flows (per-system and per-module), 4k per port
    - CES: 4k flows (L2) or 2k flows (L2/L3)
  - Support for Hybrid port
    - MLXe/XMR: 2k Protected vs 4k Unprotected VLAN IDs
    - CES: Doesn't support Hybrid Port
  - Number of controllers supported
    - MLXe/XMR and CES: 3 (active or passive)
    - SSL optional (max of two)
  - Kind of matches supported:
    - MLXe/XMR: L2, L3, L2/L3 (L2/L3 only on 8x10G and 2x100G)
    - CES: L2 and L2/L3 (L3 in future)
  - Default actions:
    - Drop packets or Send to controller
  - Statistics per Flow
    - MLXe/XMR: all
    - CES: First 2k flows
    - sFlow supported in Openflow ports

Step 3: OpenFlow Controller and Orchestrator (1/2)

- Controller vs Orchestrator
  - Controller: manages the southbound interface
    - Ex.: NOX, POX, FloodLight, OpenDayLight, RYU, etc..
  - Orchestrator: business application

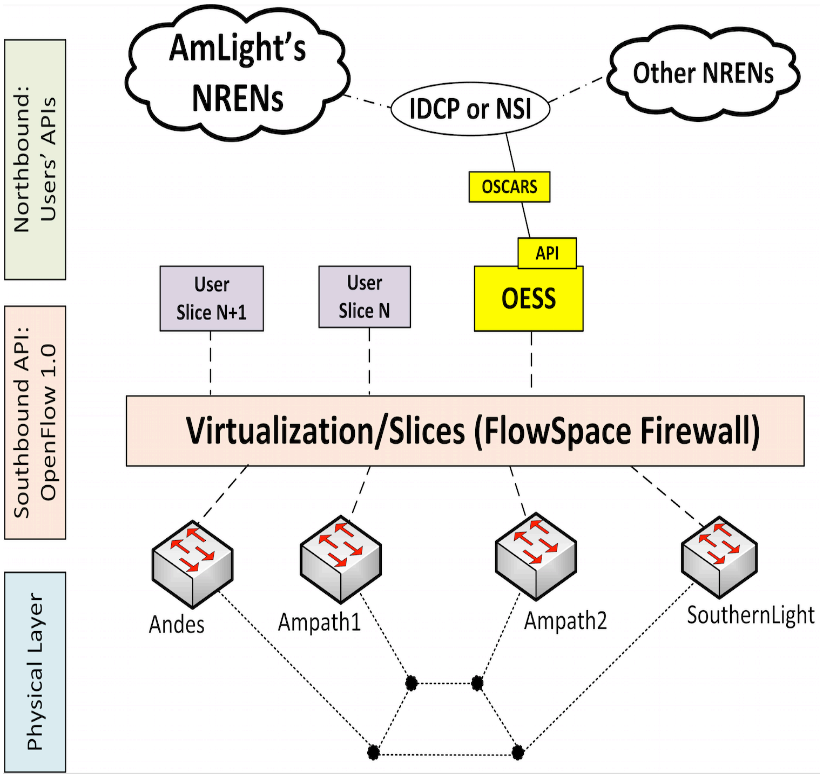
- In-house development? Use one available?
  - Does it support your applications/services in use?

- Do you need network virtualization/slicing?

Step 3: OpenFlow Controller and Orchestrator (2/2)

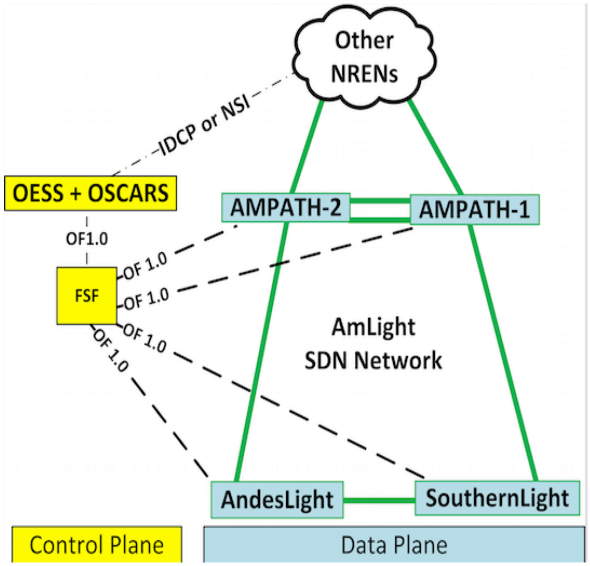
AmLight Use Case:

- OpenFlow 1.0
- Controller: NOX
- Orchestrator: Internet2 OESS
  - Supports Layer 2 provisioning via Web User Interface
  - Supports OSCARS (multi-domain provisioning – useful for RENs)
- New feature added:
  - Network Virtualization: FlowSpace Firewall



Step 4: OpenFlow control plane network

- Where to place the controller?
  - Important question for WAN, not that much for Campus/Datacenters
  - How many controllers?
    - One per site, One per domain?
- How to reach all network devices from controller?
  - In band?
    - Most vendors don’t support Openflow messages over Openflow flow entries
  - Out of Band?
    - Is there connectivity restraint? New interfaces required?
- Final configuration:
  - <u>Out of band through a third party network</u>
  - Controller installed in Miami (closed to the engineers)



Step 5: Security

- Use SSL or not for the control plane?
  - With SSL: secure communication, not all controllers support. Hard to troubleshoot (*tcpdump*)
  - Without SSL: insecure, all controllers support, easy troubleshooting

- Control the amount of flows per slice/virtualized network

- Create flow insertion rate-limit per slice/virtualized network

- How to troubleshoot in a passive way through a secure approach?

AmLight Use Case:

- Use SSL or not for the control plane?
  - Without SSL: easy troubleshooting
  - Limited to 4000 L2/L3 flows (360 in use)
  - 40 flows/sec flow insertion rate-limit (15 observed)

- How to troubleshoot in a passive way through a secure approach?
  - OpenFlow Sniffer developed!

Step 6: Deployment

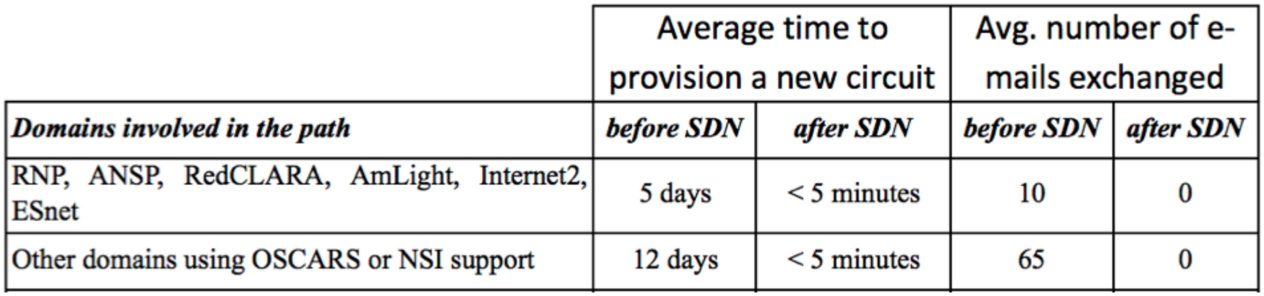
- New skills required: Linux, Log Reading, Coding (Python or Java), etc.
- Start with *mininet*, try to reproduce your services and configurations
- Create a testing environment with real switches and, if possible, the same vendors and models
- If possible, deploy it gradually, for example, start with layer 2 services

AmLight Use Case:

- Started on April 30th, Deployed on August 30th
- A few tools developed for troubleshooting
- New skills acquired: OpenFlow and Python (Linux was part of our routine)
- Peaks of 55Gbps of traffic observed
  - SuperComputing 2014

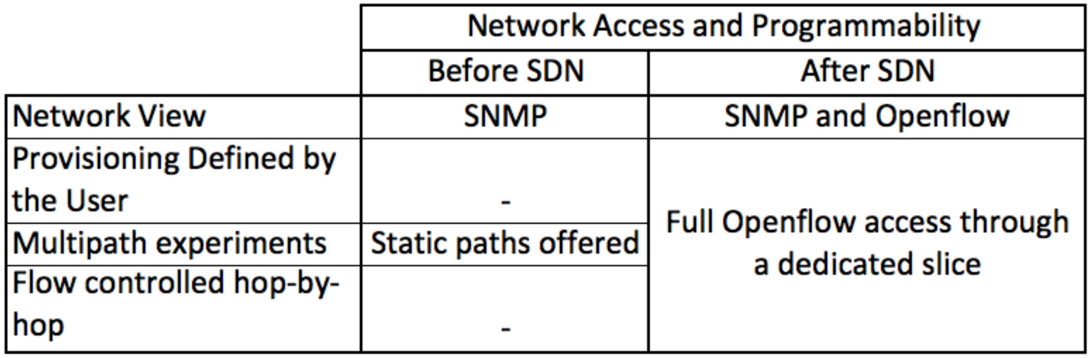
Results

Provisioning:



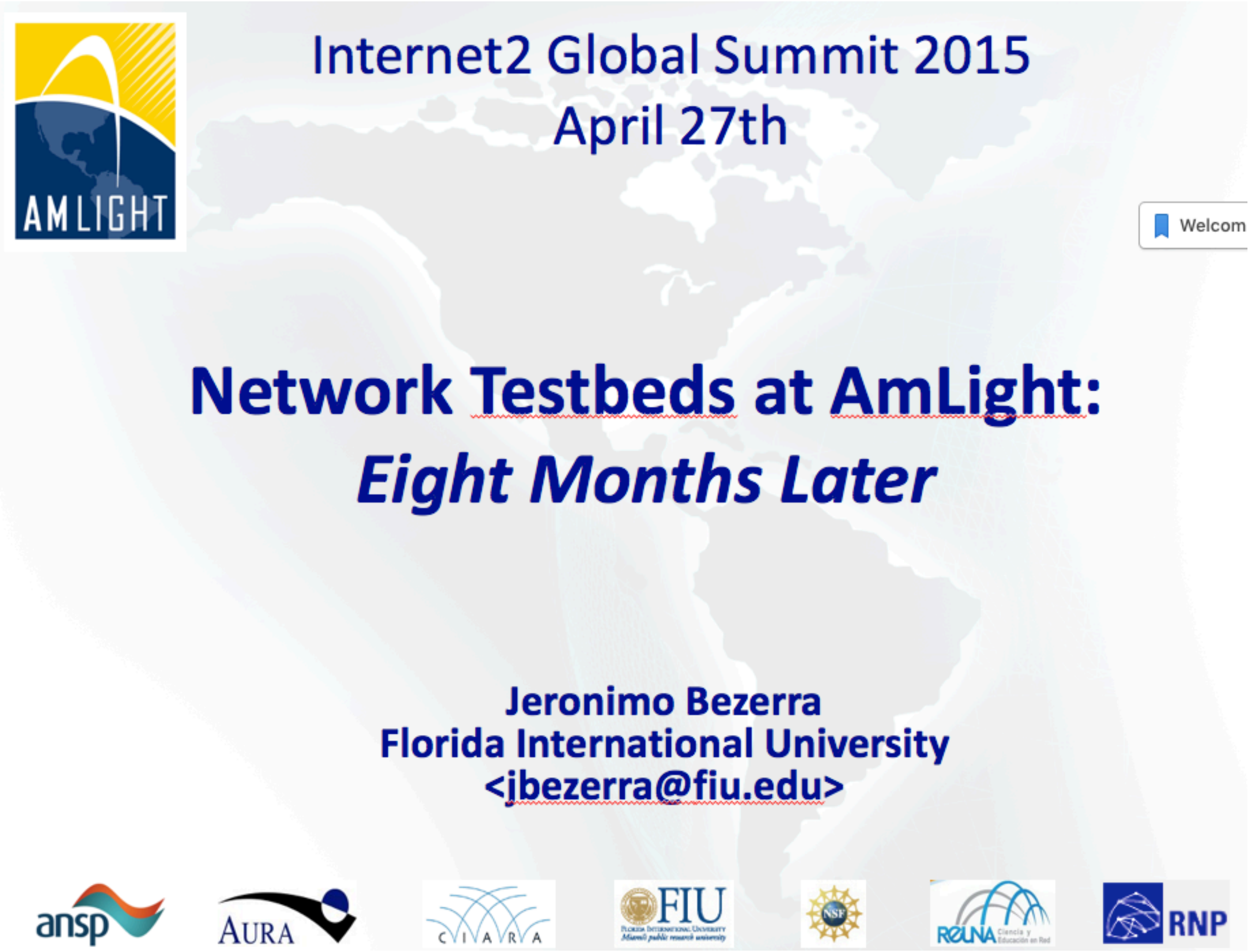
	Average time to provision a new circuit		Avg. number of e-mails exchanged	
Domains involved in the path	***before SDN***	***after SDN***	***before SDN***	***after SDN***
RNP, ANSP, RedCLARA, AmLight, Internet2, ESnet	5 days	< 5 minutes	10	0
Other domains using OSCARS or NSI support	12 days	< 5 minutes	65	0

Programmability:



	Network Access and Programmability	
	Before SDN	After SDN
Network View	SNMP	SNMP and Openflow
Provisioning Defined by the User	-	Full Openflow access through a dedicated slice
Multipath experiments	Static paths offered	
Flow controlled hop-by-hop	-	

Some months later…



Internet2 Global Summit 2015
April 27th

Network Testbeds at AmLight:
Eight Months Later

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

AMLIGHT
Welcom
ansp
AURA
CIARA
FIU
Florida International University
Miami's public research university
NSF
REUNA Ciencia y Educación en Red
RNP

Network Testbeds at AmLight SDN

Network Testbeds offered through Network Slices:

- Network Slices:
    - Defined by a set of Interfaces and VLANs
    - Each Slice has its own Openflow Controller
    - Different Topologies Available

- How does AmLight support slices?
    - Internet2 Flow Space Firewall (FSF) is being used to create slices
    - FSF talks OpenFlow 1.0 to controller and network devices
    - Provides isolation between slices
    - Filters OpenFlow messages based on Interfaces and VLANs
    - Support filters: # of flows inserted and flows inserted per second.
    - Supports a high # of parallel slices

Who is using AmLight SDN?

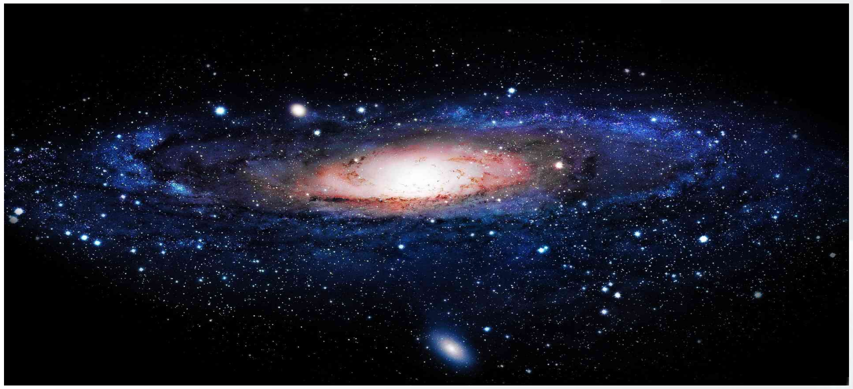
Current Testbeds (1/2)

- **FIBRE: Interconnecting Testbed's Islands with OpenFlow**
- **NSI testing deployment**
- **Testing new controllers and applications in a separated slice**
- **OpenFlow Statistics Validation**
  - PhD study at the University of Twente, The Netherlands
- **Demonstrations**
  - Internet2 Multi-Domain Slices (Oct 2014 I2 Tech Exchange Meeting)
  - Internet2 Inter-Domain IP connections (Apr 2015 I2 Global Summit)
  - ONOS Global Deployment (May 2016)

Eight Months Later: Lessons Learned

Each new Network Testbed is a new challenge: new apps, new methodology and always complex!

Researchers expectations:



"Need" full access to everything!

Requires a lot of singularities:

Untagged VLANs, Reactive Openflow Mode, Specific Actions, Specific Matches, Direct Access to the Openflow devices, …

AmLight possibilities:

It's a Shared Environment!

Complexity involved for "big" changes:

Proactive Mode, Untagged VLAN, etc.

Main Challenge Today is to Balance Expectations!
We should avoid more obstacles to researchers!

13 Months Later

Internet2 Technology Exchange 2015
Oct 06th

Coexisting Production and Experimental Testbeds at AmLight: A Secure Approach

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

Marcos Schwarz
Rede Nacional de Ensino e Pesquisa
<marcos.schwarz@rnp.br>

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

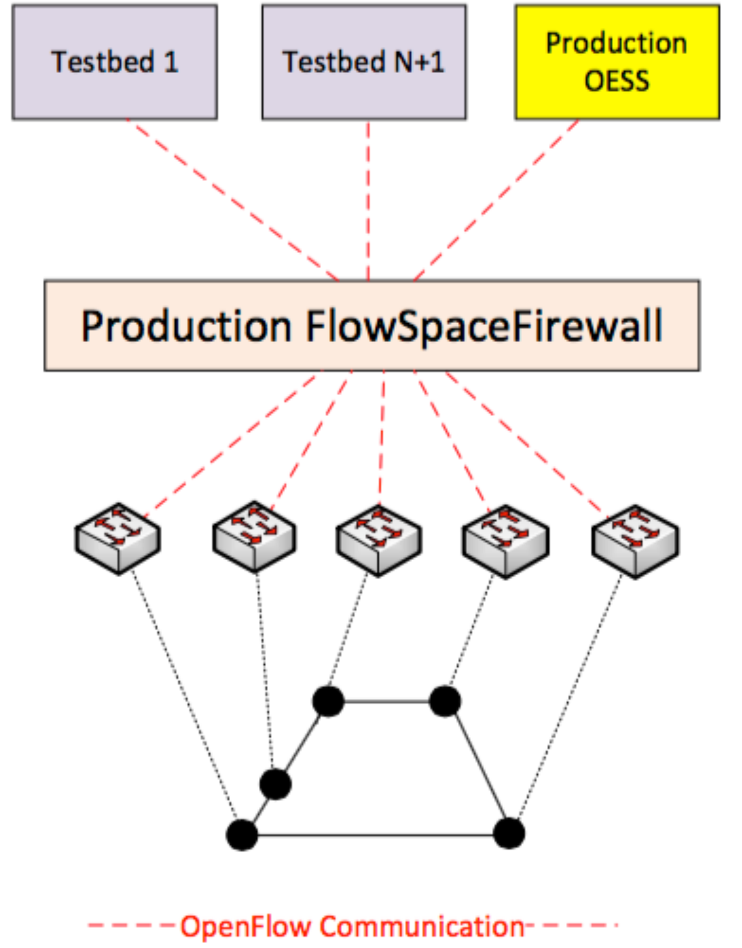
- Multiple OF controllers could manage the same OpenFlow device:
  - Complicated to isolate who is sending specific OF messages

- 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 crashes happened, hard to understand "why"



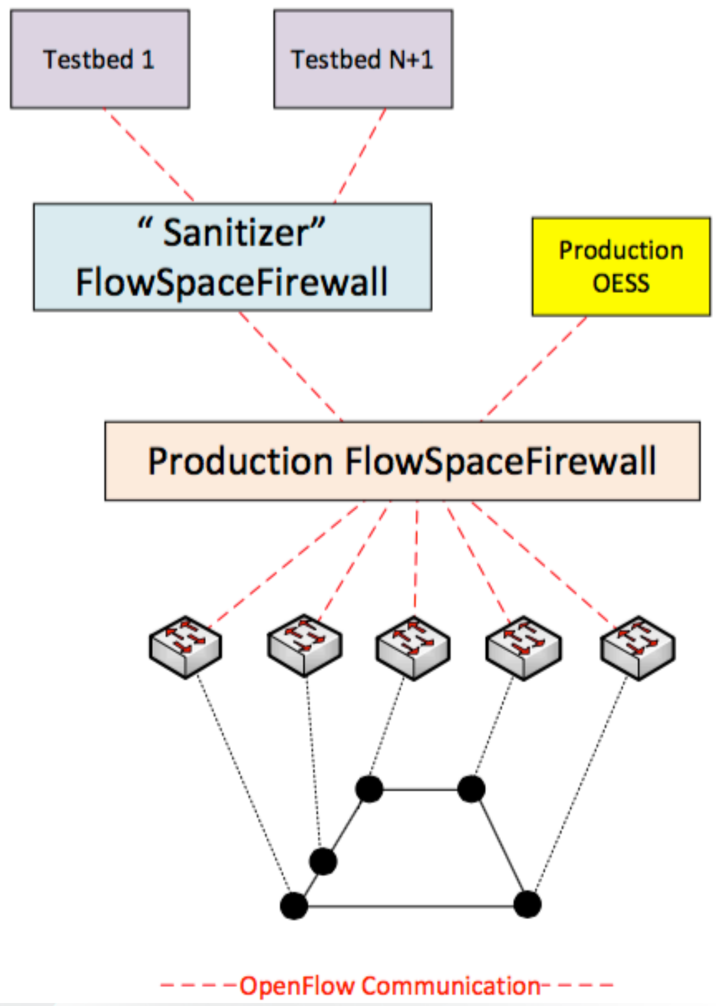
New Architecture – Proof of Concept

- 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?)

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.

Internet2 Technology Exchange
SDN-WG/The Exchange
Oct 5/6th 2015

SDN tools at AmLight: an OpenFlow sniffer

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

We needed troubleshooting tools: OFP_Sniffer

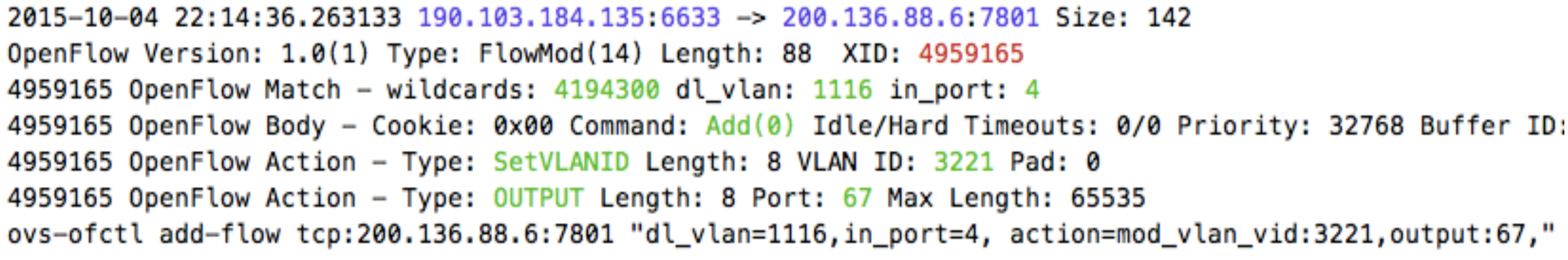
Motivation

- 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)

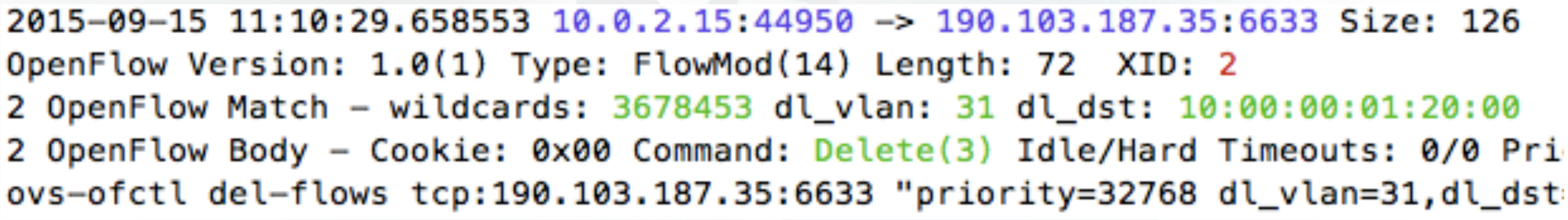
Features

- OpenFlow 1.0 support
- Runs on Linux shell
  - No need for X Windows
- Supports all OpenFlow 1.0 messages (1.3 almost completed)
- Highlights 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

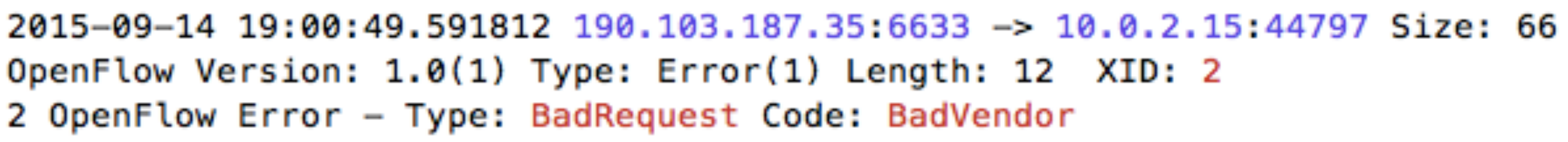
Outputs



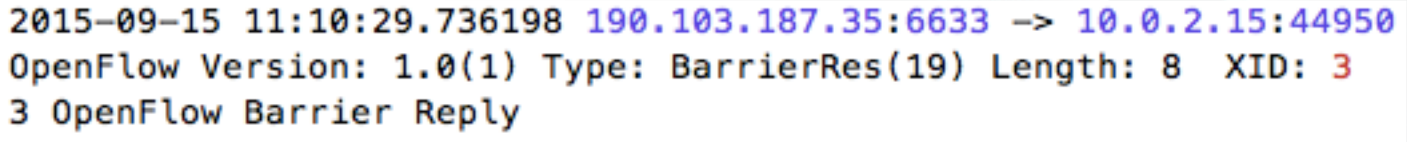
```
2015-10-04 22:14:36.263133 190.103.184.135:6633 -> 200.136.88.6:7801 Size: 142
OpenFlow Version: 1.0(1) Type: FlowMod(14) Length: 88  XID: 4959165
4959165 OpenFlow Match - wildcards: 4194300 dl_vlan: 1116 in_port: 4
4959165 OpenFlow Body - Cookie: 0x00 Command: Add(0) Idle/Hard Timeouts: 0/0 Priority: 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
ovs-ofctl add-flow tcp:200.136.88.6:7801 "dl_vlan=1116,in_port=4, action=mod_vlan_vid:3221,output:67,"
```



```
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 Pri
ovs-ofctl del-flows tcp:190.103.187.35:6633 "priority=32768 dl_vlan=31,dl_dst
```

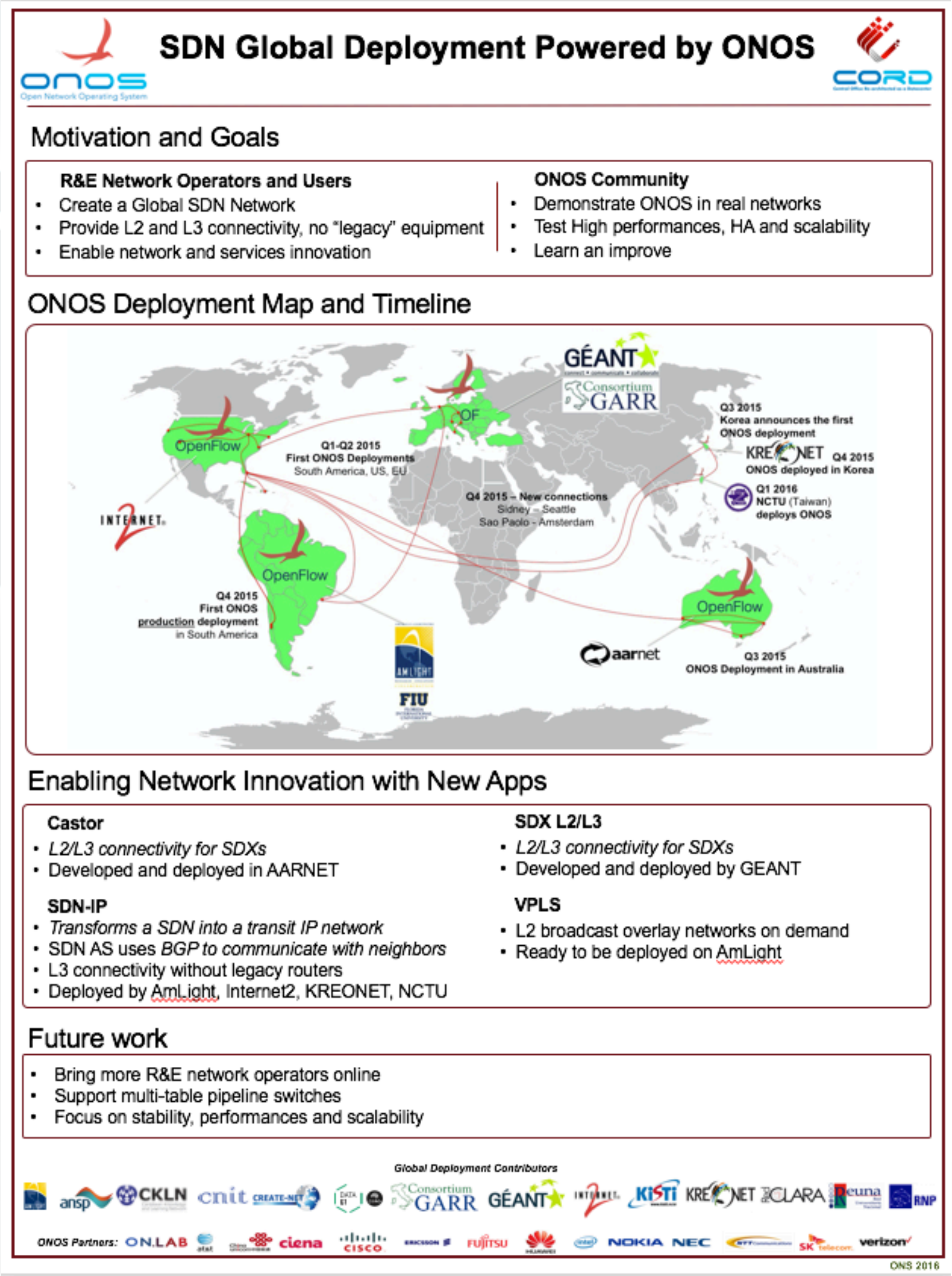


```
2015-09-14 19:00:49.591812 190.103.187.35:6633 -> 10.0.2.15:44797 Size: 66
OpenFlow Version: 1.0(1) Type: Error(1) Length: 12  XID: 2
2 OpenFlow Error - Type: BadRequest Code: BadVendor
```



```
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
```

ONOS?



SDN Global Deployment Powered by ONOS

Motivation and Goals

R&E Network Operators and Users

- Create a Global SDN Network
- Provide L2 and L3 connectivity, no "legacy" equipment
- Enable network and services innovation

ONOS Community

- Demonstrate ONOS in real networks
- Test High performances, HA and scalability
- Learn an improve

ONOS Deployment Map and Timeline

OpenFlow
OF
OpenFlow
OpenFlow
Q1-Q2 2015
First ONOS Deployments
South America, US, EU
Q3 2015
Korea announces the first
ONOS deployment
Q4 2015
ONOS deployed in Korea
Q1 2016
NCTU (Taiwan)
deploys ONOS
Q4 2015 – New connections
Sidney – Seattle
Sao Paolo - Amsterdam
Q4 2015
First ONOS
production deployment
in South America
Q3 2015
ONOS Deployment in Australia

Enabling Network Innovation with New Apps

Castor

- *L2/L3 connectivity for SDXs*
- Developed and deployed in AARNET

SDN-IP

- *Transforms a SDN into a transit IP network*
- SDN AS uses *BGP to communicate with neighbors*
- L3 connectivity without legacy routers
- Deployed by AmLight, Internet2, KREONET, NCTU

SDX L2/L3

- *L2/L3 connectivity for SDXs*
- Developed and deployed by GEANT

VPLS

- L2 broadcast overlay networks on demand
- Ready to be deployed on AmLight

Future work

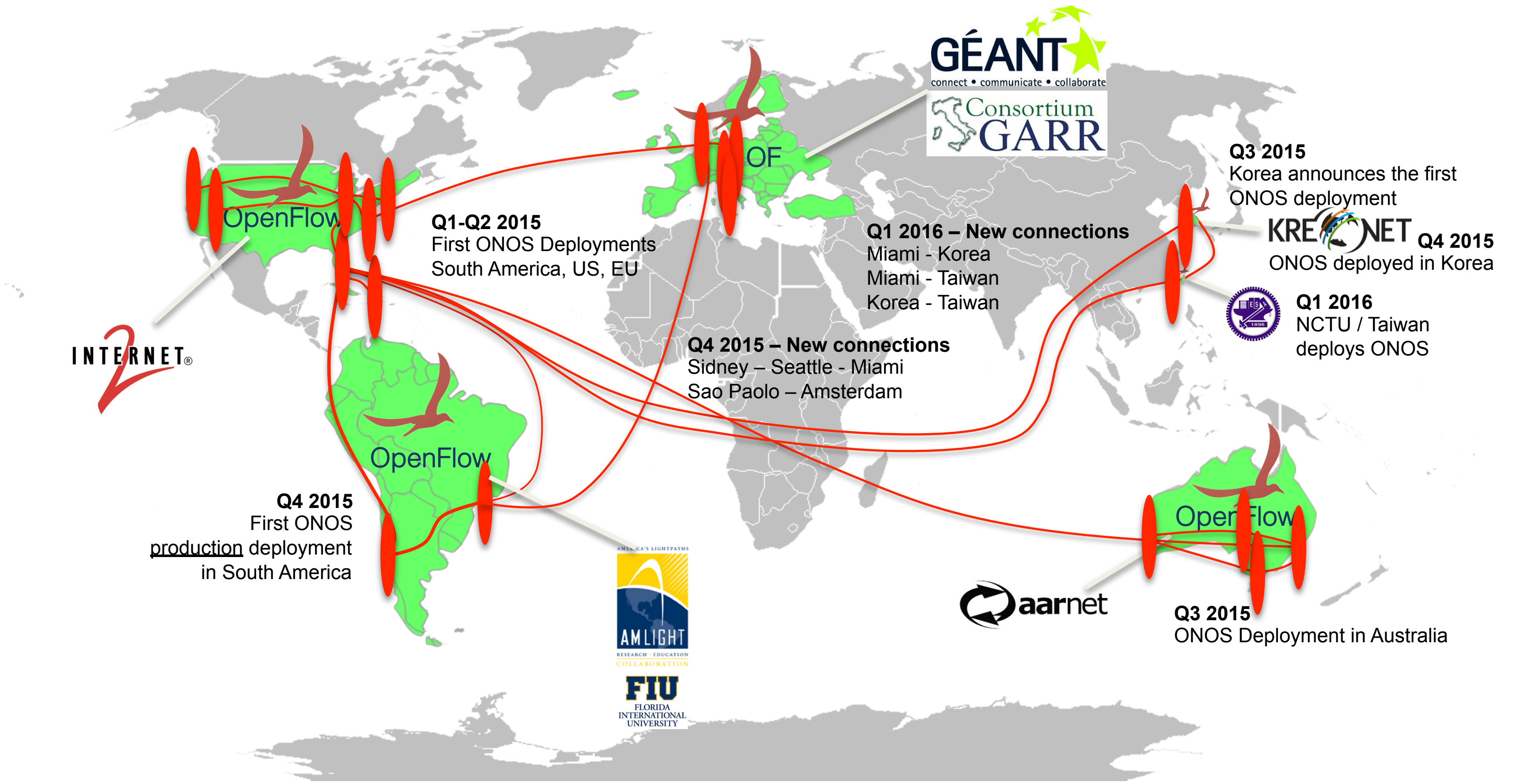
- Bring more R&E network operators online
- Support multi-table pipeline switches
- Focus on stability, performances and scalability

Global Deployment Contributors

ONOS Partners:

ONS 2016

Global SDN Deployment Powered by ONOS



Q1-Q2 2015
First ONOS Deployments
South America, US, EU

Q3 2015
Korea announces the first
ONOS deployment

Q4 2015
ONOS deployed in Korea

Q1 2016 – New connections
Miami - Korea
Miami - Taiwan
Korea - Taiwan

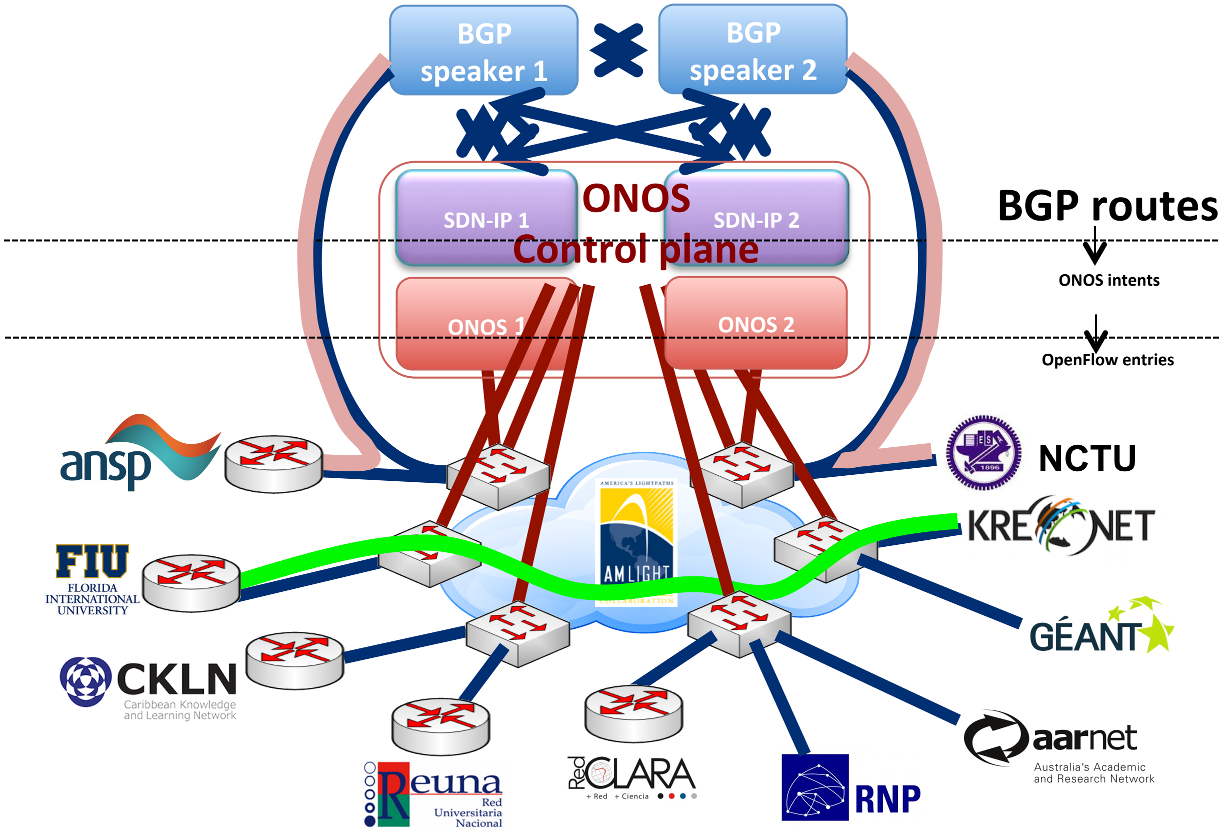
Q1 2016
NCTU / Taiwan
deploys ONOS

Q4 2015 – New connections
Sidney – Seattle - Miami
Sao Paolo – Amsterdam

Q4 2015
First ONOS
production deployment
in South America

Q3 2015
ONOS Deployment in Australia

The AmLight use case



#ONOSProject

Lessons Learned

1. SDN and OpenFlow have the potential of helping in many ISP activities
   - Especially multi-domain provisioning

2. Caution before deploying SDN applications into production
   - Gaps in troubleshooting, implementation of OpenFlow protocol, etc.
   - Many opportunities to those interested in new technologies
   - Most of the OpenFlow switches tested had immature code
   - Many features missing, poor code development, lack of tests before releasing code, etc.

Lessons Learned [2]

3. Troubleshooting is very time-consuming and frustrating
   - In most of the cases, vendors lacked proper tools for troubleshooting
   - New tools are needed urgently

4. Immature OpenFlow code = risks to production network services
   - Devices crashed due to unsupported OpenFlow messages
   - Validating OpenFlow software is very important

5. Slicing proved to be very useful for testing different control planes
   - Useful in revealing potential risks before deploying into production
   - Encourage researchers to use slicing as a testing methodology

Change of Culture @ AmLight

- From Network Engineering to SDN/Research Engineering?
  - Agile/SCRUM, Python, Java, *unittest* are part of daily discussions of the network engineering group

- Training on Software Development (programming languages, processes, etc.)
  - Python for Network Engineers was just the beginning

- Strong focus on software instead of hardware
  - Team was trained to follow the Agile/Scrum methodology
  - AmLight started to collaborate with the ONOS development (ONOS Brigade)

Network Engineer 2.0 (?)

- Main challenge was (and still is):
  - How to convince some network engineers (not managers!) that SDN “might” be useful
  - It is difficult to learn a new paradigm

- Once convinced, training, training, training!
  - Going back to school was pretty hard for some of them
  - FIU has full access to Lynda.com – easy access to great courses!

- Identity “crisis”: *what am I now: a network engineer, DevOps, SDN/Research engineer?*
  - *Maybe Network Engineer 2.0?*
  - Recruiting *Network Engineer 2.0* people is challenging

Conclusion

- Deploying SDN pushed AmLight to *reinvent* itself
  - We continue learning how to operate SDN in a production environment
  - In some cases, CAPEX increased, while OPEX kept the same or decreased
  - Troubleshooting is still complex and time-consuming, but it is part of the game

- Facilitating development of SDN applications
  - Validation methodology for the controller
  - Bring us your testbeds

- After two years, we consider the investment in SDN a success
  - Looking forward to enhance collaboration with other R&E networks

Questions?

Jerônimo Aguiar Bezerra
<jbezerra@fiu.edu>

Goals for the future

- Migrate to OpenFlow 1.3
  - Running some experiments with Corsa and other vendors

- New tools: SDN Looking Glass
  - Unified GUI interface to operate OpenFlow networks
  - Independent of slicers/virtualization: full network visualization
    - Statistics
    - Passive and Active tests
    - Real time monitoring
  - Independent of OpenFlow controller: Ryu, ODL and ONOS supported
  - ETD: February 2017