2023 WTR BA

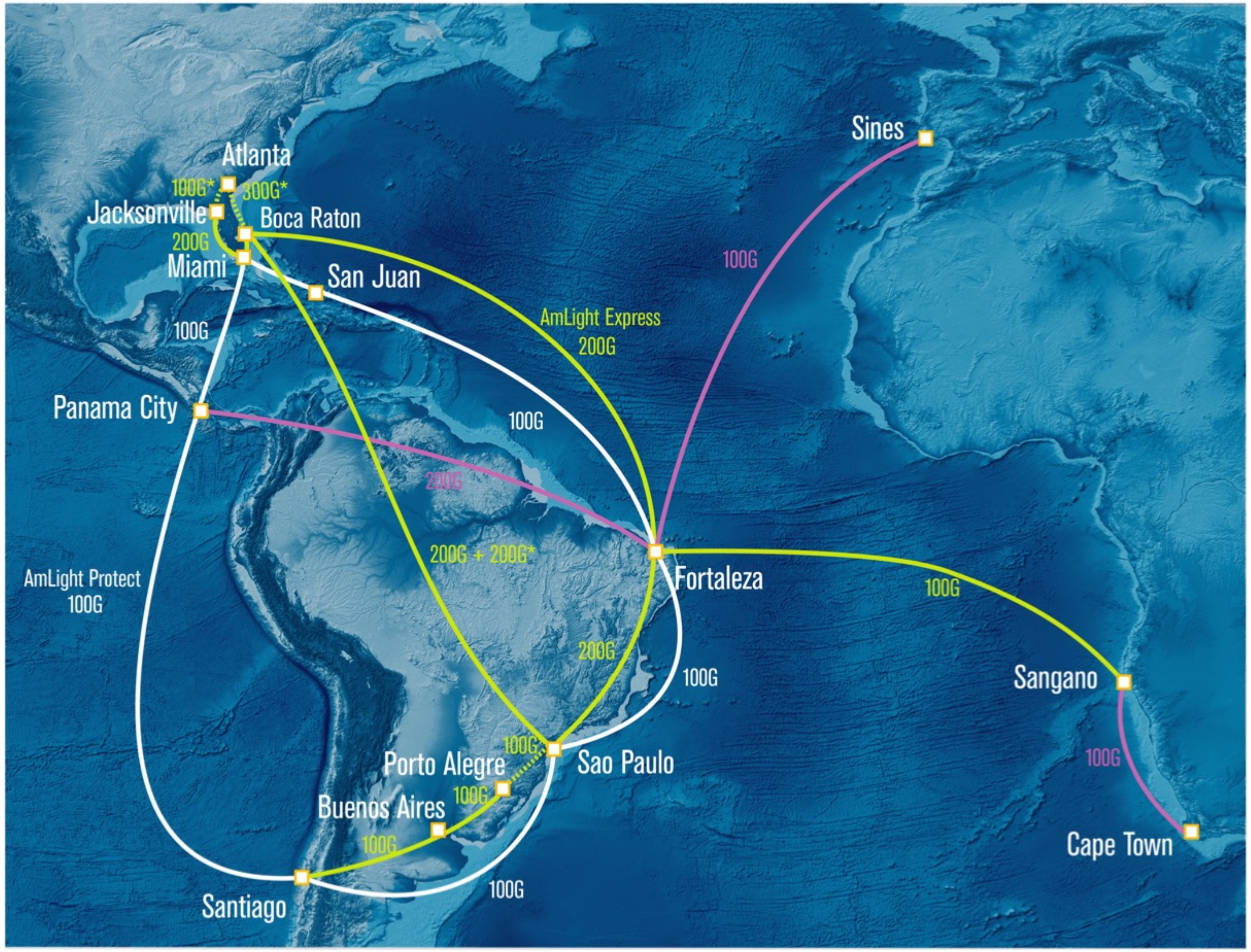
Kytos-ng at AmLight: the new generation of SDN orchestrator

Italo Valcy, Senior Network Engineer

italo@amlight.net

AmLight-ExP Network Infrastructure

- 600G of upstream capacity between the U.S., Latin America, Caribbean and 100G to Africa
  - Blend of Spectrum and Leased Capacity
- OXPs: Florida(3), Brazil(2), Chile, Puerto Rico, Panama, and South Africa, (soon Atlanta, GA)
- Production SDN Infrastructure since 2014:
  - Orchestrators: OESS and Kytos-ng
  - OpenFlow 1.0 and 1.3 Southbound Interfaces
- Programmable Data Plane:
  - P4 -> In-band Network Telemetry (INT)
  - 21 programmable devices in production
- Highly instrumented:
  - PerfSonar, sFlow, Juniper Telemetry Interface (JTI), In-band Network Telemetry (INT)

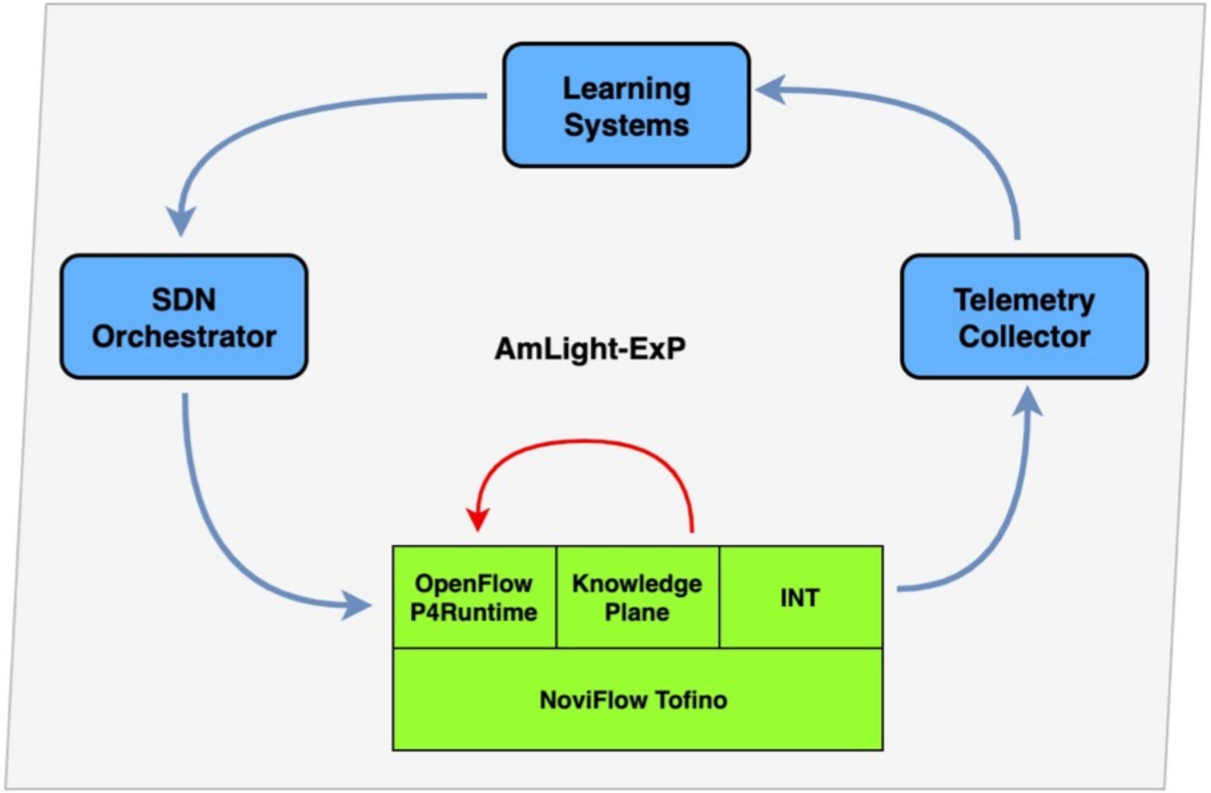


SDN Orchestrator challenges

- Major Goal: Improving Resiliency, Increasing Flexibility and Self-Management
    - SDN Orchestrator plays an important role

- Providing a Closed-Loop Orchestration

- The new SDN orchestrator should support AmLight network operation:
    - User requirements not easily supported by “commodity” NOS and SDN Controllers
    - Pathfinding application handling unusual metrics (reliability, ownership, max delay)
    - Granular network telemetry per flows, per protocols, etc
    - Increased capacity to innovate
    - Critical infrastructure



Kytos-ng

- The AmLight SDN Orchestrator is built over the Kytos-ng SDN Platform
  - https://kytos-ng.github.io

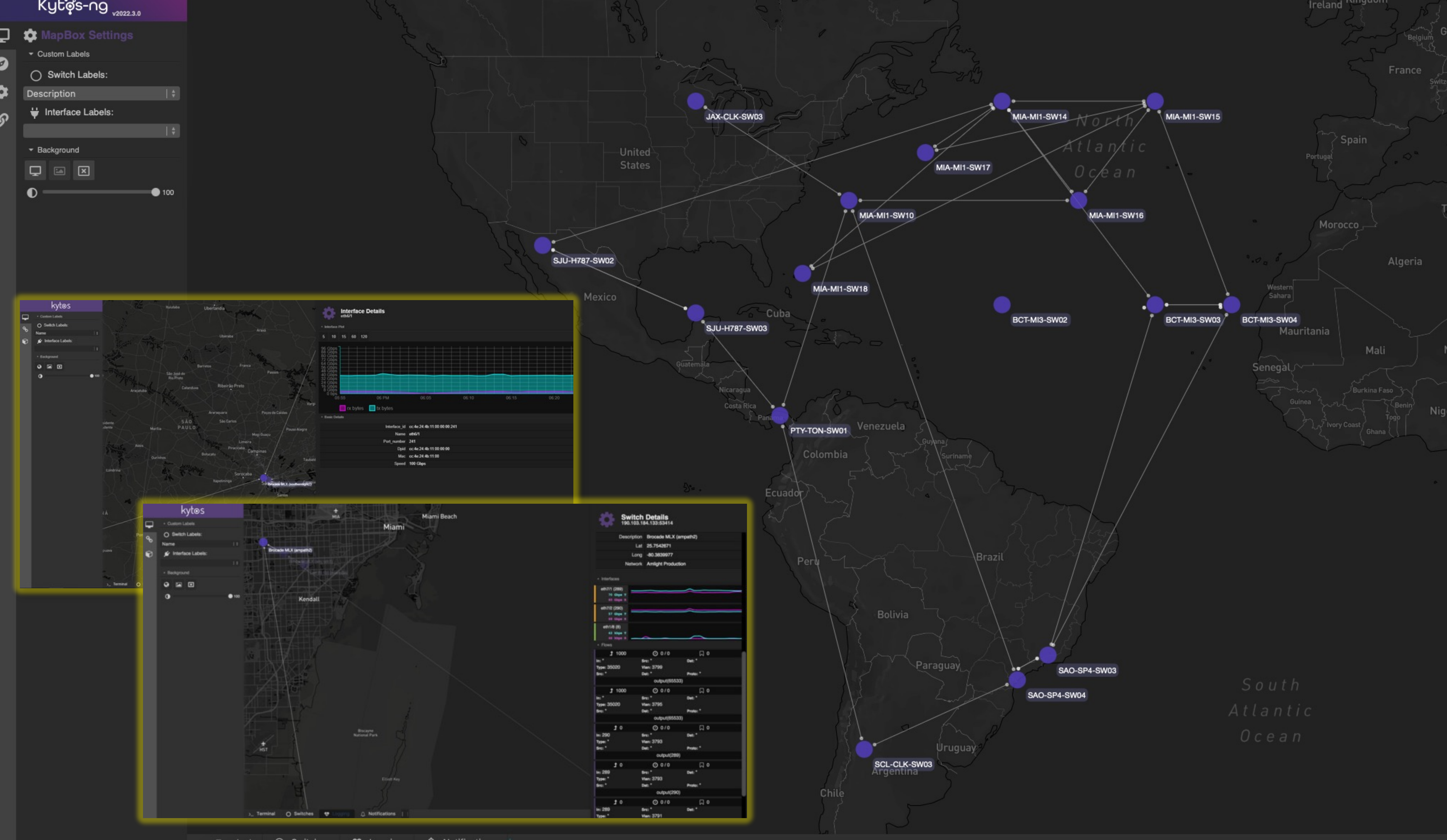
- Python application designed to be very powerful and scalable, with a lightweight core and Network Applications (Napps) as extensions
  - https://github.com/orgs/kytos-ng/repositories
  - https://napps.kytos.io

- Kytos-ng SDN Platform is possible thanks to the support provided by rednesp (Research and Education Network of Sao Paulo), AMPATH/AmLight team, and by the Kytos open source community

AmLight SDN Orchestrator new generation

- Kytos-ng is deployed as AmLight SDN Control Plane since 2022 (along with OESS), current release 2022.3.3 (moving towards 2023.1)
    - A new instance of the CP running from Miami, orchestrating Noviflow switches in Jacksonville, Miami, Boca Raton, Sao Paulo, Santiago, and Panama City

- Kytos-ng in numbers:
    - 15 Switches
    - 24 100Gb Links
    - 164 EVCs
    - 1651 installed flows



Kytos-ng v2022.3.0

MapBox Settings

Custom Labels

○ Switch Labels: Description

Interface Labels:

Background

100

Interface Details
eth6/1

Basic Details

Interface_id cc:4e:24:4b:11:00:00:00:241
Name eth6/1
Port_number 241
Dpid cc:4e:24:4b:11:00:00:00
Mac cc:4e:24:4b:11:00
Speed 100 Gbps

Switch Details
190.103.184.133:53414

Description Brocade MLX (ampath2)
Lat 25.7542671
Long -80.3839977
Network Amlight Production

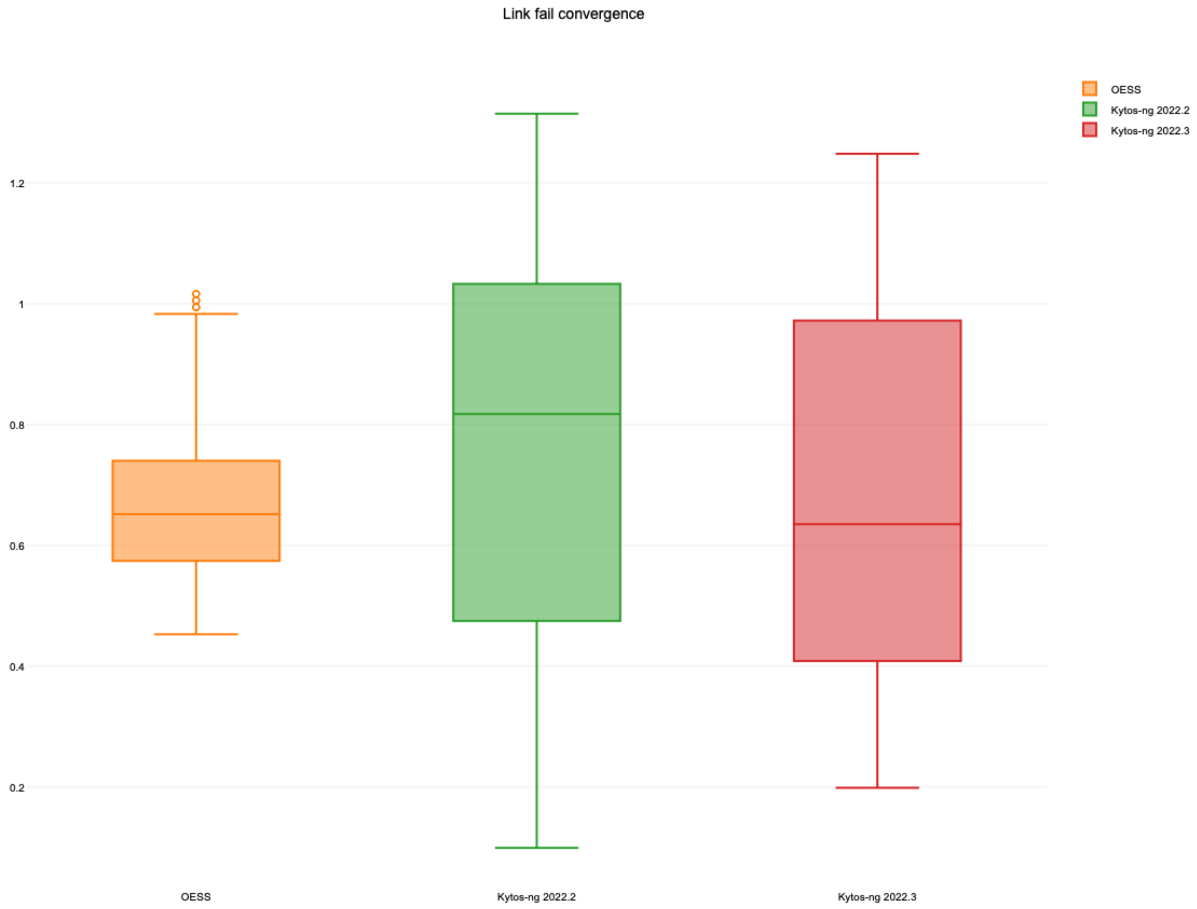
Kytos-ng key features

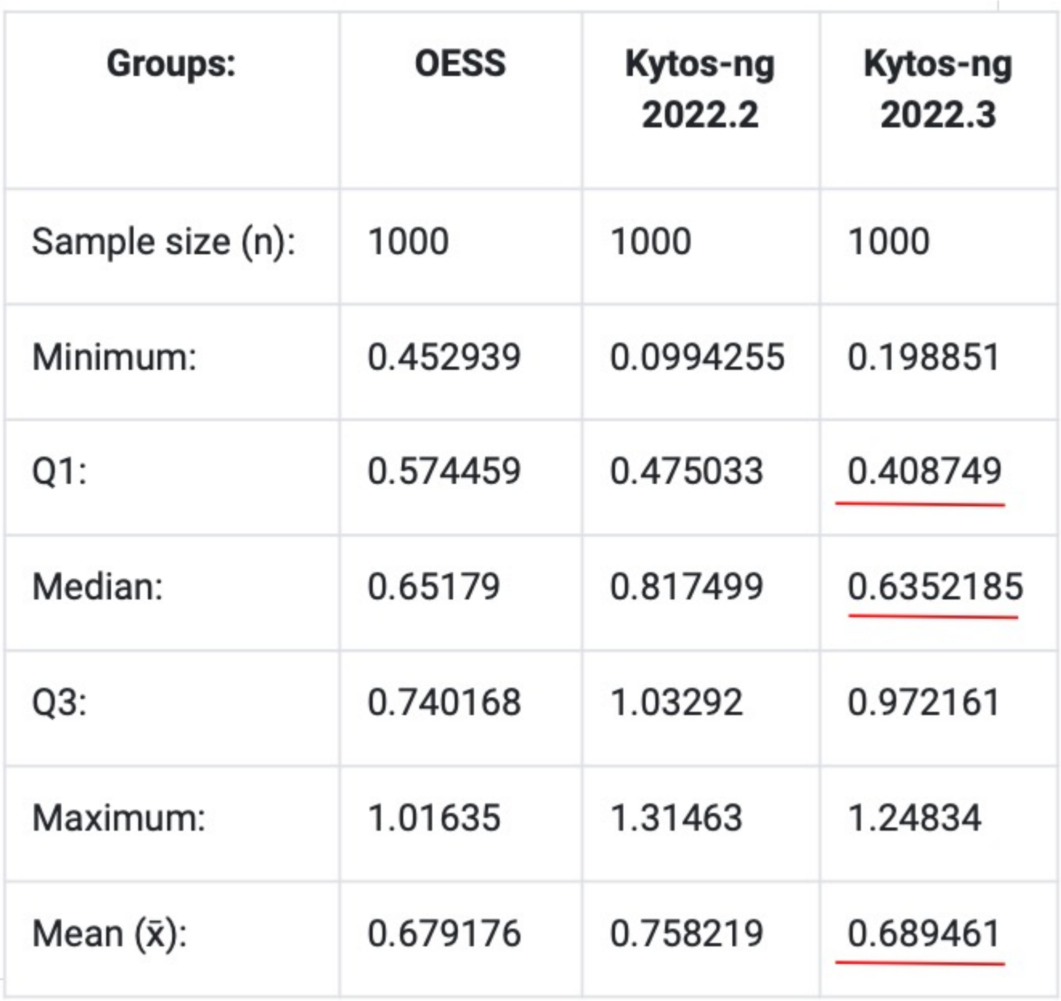
- MEF-compliance point-to-point L2VPN
- Enriched Path Finding strategy (best, constraint, policy-based, etc)
- Topology discovery, manager and visualization
- Link liveness check and forwarding verification
- Path tracing (control plane and data plane)
- Maintenance Tasks helper
- Multi-level application/network consistency check
- Web UI and REST API, persistency, micro-service, event based, async

MEF E-line ptp L2VPN

- Fully compliance with MEF E-line spec
- EPL / EVPL
- Static, Dynamic, Constraint paths
- Fast Failover
- Hop-by-hop QinQ
- VLAN range, QoS

Performance tests - Link failure convergence





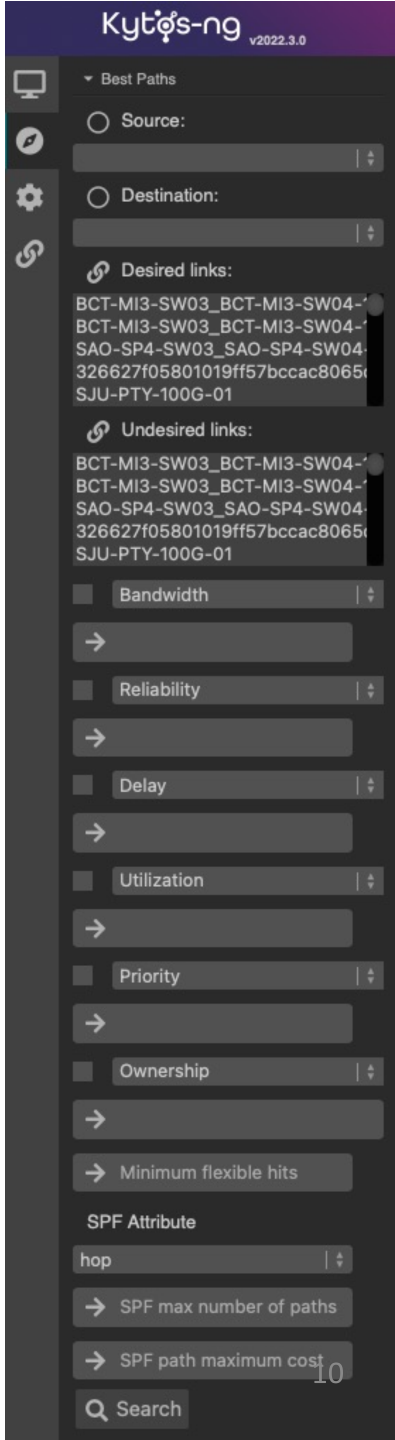
Groups:	OESS	Kytos-ng 2022.2	Kytos-ng 2022.3
Sample size (n):	1000	1000	1000
Minimum:	0.452939	0.0994255	0.198851
Q1:	0.574459	0.475033	0.408749
Median:	0.65179	0.817499	0.6352185
Q3:	0.740168	1.03292	0.972161
Maximum:	1.01635	1.31463	1.24834
Mean (x̄):	0.679176	0.758219	0.689461

Enriched Path Finding

- Kytos-ng has a vastly customizable and enriched path finding strategy
  - The networks is modeled as an undirected graph
  - Link metadata are enriched by native Kytos stats and external data source

- Supports additional filtering capabilities and CSPF (Constraint SPF)
  - Flexible requirements are always evolving
  - Working in progress to support Maximum Disjoint Paths (aware of shared resource)

- Unique level of TE, difficult to find in other solutions (SDN and legacy)



Kytos-ng v2022.3.0

Best Paths

○ Source:

○ Destination:

Desired links:

BCT-MI3-SW03_BCT-MI3-SW04-
BCT-MI3-SW03_BCT-MI3-SW04-
SAO-SP4-SW03_SAO-SP4-SW04
326627f05801019ff57bccac8065
SJU-PTY-100G-01

Undesired links:

BCT-MI3-SW03_BCT-MI3-SW04-
BCT-MI3-SW03_BCT-MI3-SW04-
SAO-SP4-SW03_SAO-SP4-SW04
326627f05801019ff57bccac8065
SJU-PTY-100G-01

□ Bandwidth

□ Reliability

□ Delay

□ Utilization

□ Priority

□ Ownership

Minimum flexible hits

SPF Attribute

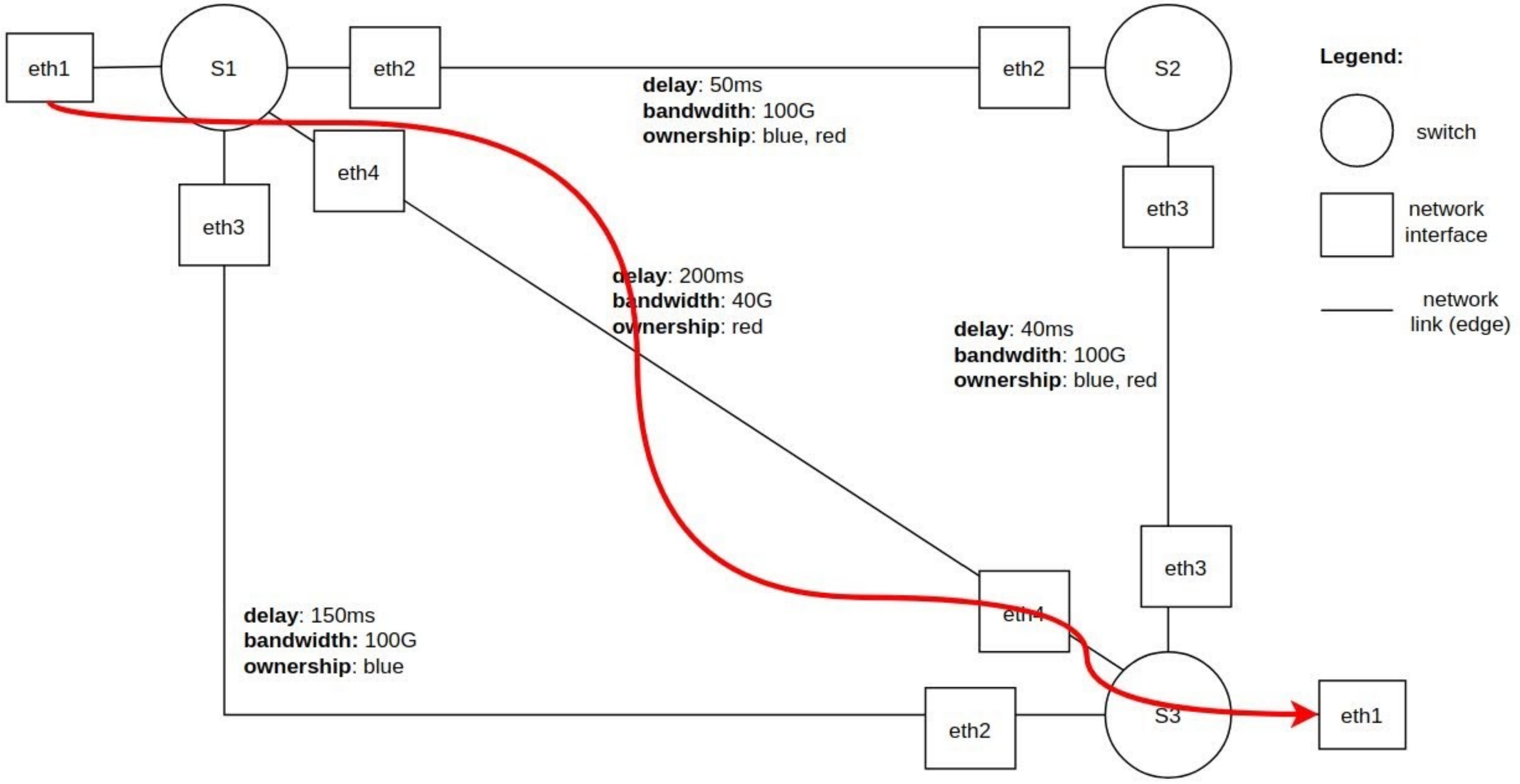
hop

SPF max number of paths

SPF path maximum cost

Search

Enriched Path Finding

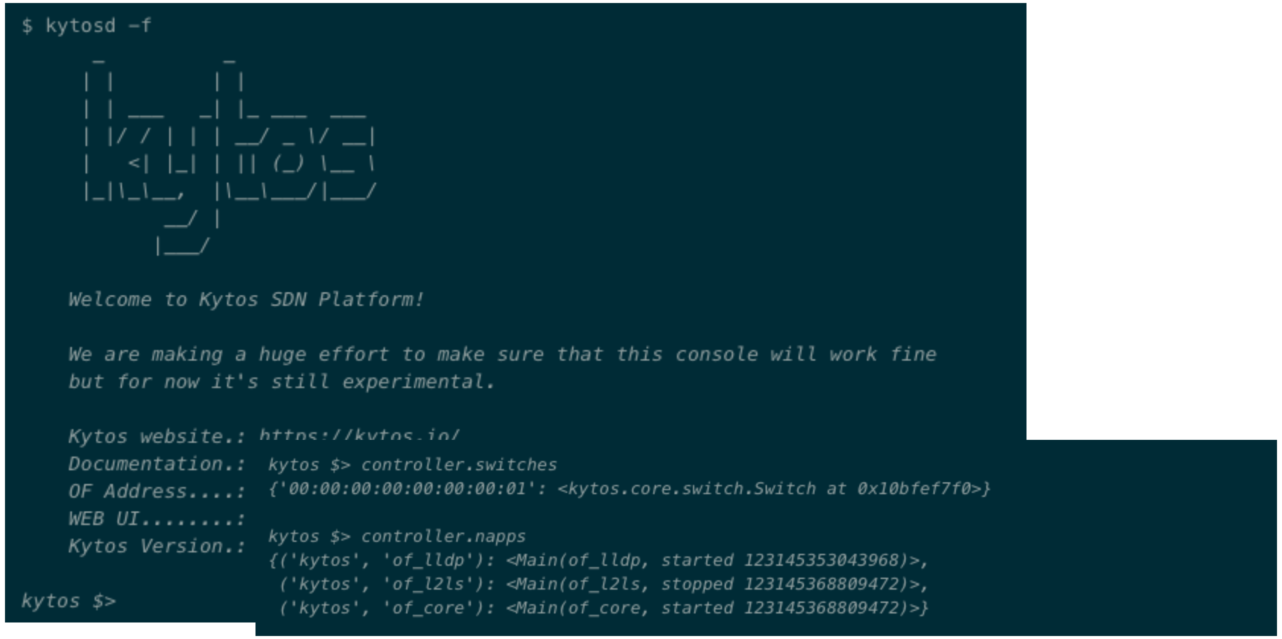


- source: S1:eth1
- destination: S3:eth1
- mandatory filters: ownership: red
- flexible filters: bandwidth: 200 Gbps, delay: 250 ms
- minimum flexible hits: 1

https://github.com/kytos-ng/kytos/blob/master/docs/blueprints/EP023-2.rst

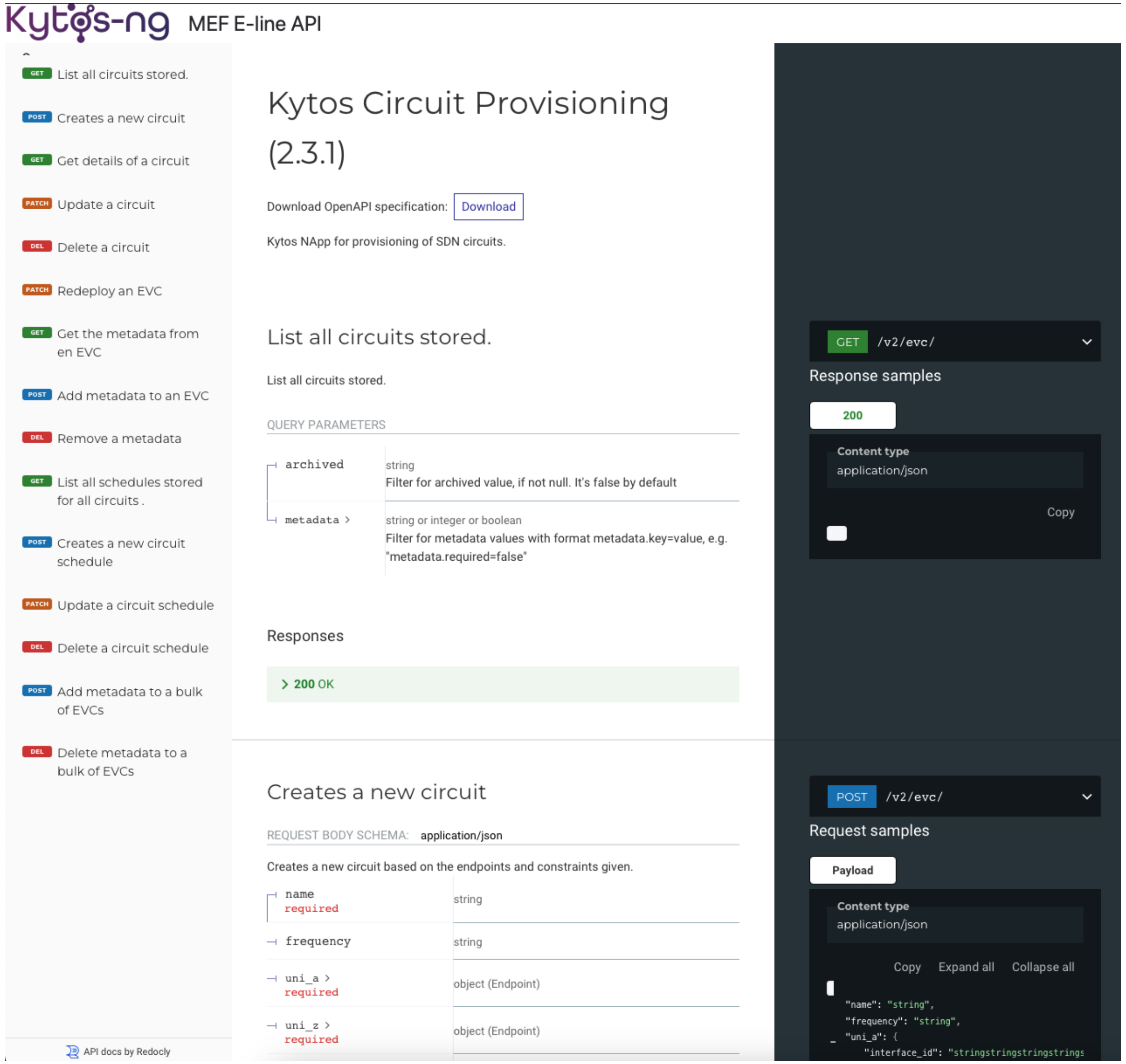
Kytos-ng user interfaces

- Web UI
- REST API
  - evc-manager
- Console



```
$ kytosd -f
      _          _
     | |        | |
     | | ___   _| |_ ___  ___
     | |/ / | | | __/ _ \/ __|
     |   <| |_| | || (_) \__ \
     |_|\_\\__, |\__\___/|___/
            __/ |
           |___/

    Welcome to Kytos SDN Platform!

    We are making a huge effort to make sure that this console will work fine
    but for now it's still experimental.

    Kytos website.: https://kytos.io/
    Documentation.:
    OF Address....:
    WEB UI........:
    Kytos Version.:

kytos $>
```

```
kytos $> controller.switches
{'00:00:00:00:00:00:00:01': <kytos.core.switch.Switch at 0x10bfef7f0>}

kytos $> controller.napps
{('kytos', 'of_lldp'): <Main(of_lldp, started 123145353043968)>,
 ('kytos', 'of_l2ls'): <Main(of_l2ls, stopped 123145368809472)>,
 ('kytos', 'of_core'): <Main(of_core, started 123145368809472)>}
```



Kytos-ng MEF E-line API

GET List all circuits stored.
POST Creates a new circuit
GET Get details of a circuit
PATCH Update a circuit
DEL Delete a circuit
PATCH Redeploy an EVC
GET Get the metadata from en EVC
POST Add metadata to an EVC
DEL Remove a metadata
GET List all schedules stored for all circuits .
POST Creates a new circuit schedule
PATCH Update a circuit schedule
DEL Delete a circuit schedule
POST Add metadata to a bulk of EVCs
DEL Delete metadata to a bulk of EVCs
API docs by Redocly

Kytos Circuit Provisioning (2.3.1)

Download OpenAPI specification: Download

Kytos NApp for provisioning of SDN circuits.

List all circuits stored.

List all circuits stored.

QUERY PARAMETERS

archived	string Filter for archived value, if not null. It's false by default
metadata ›	string or integer or boolean Filter for metadata values with format metadata.key=value, e.g. "metadata.required=false"

Responses

› 200 OK

Creates a new circuit

REQUEST BODY SCHEMA: application/json

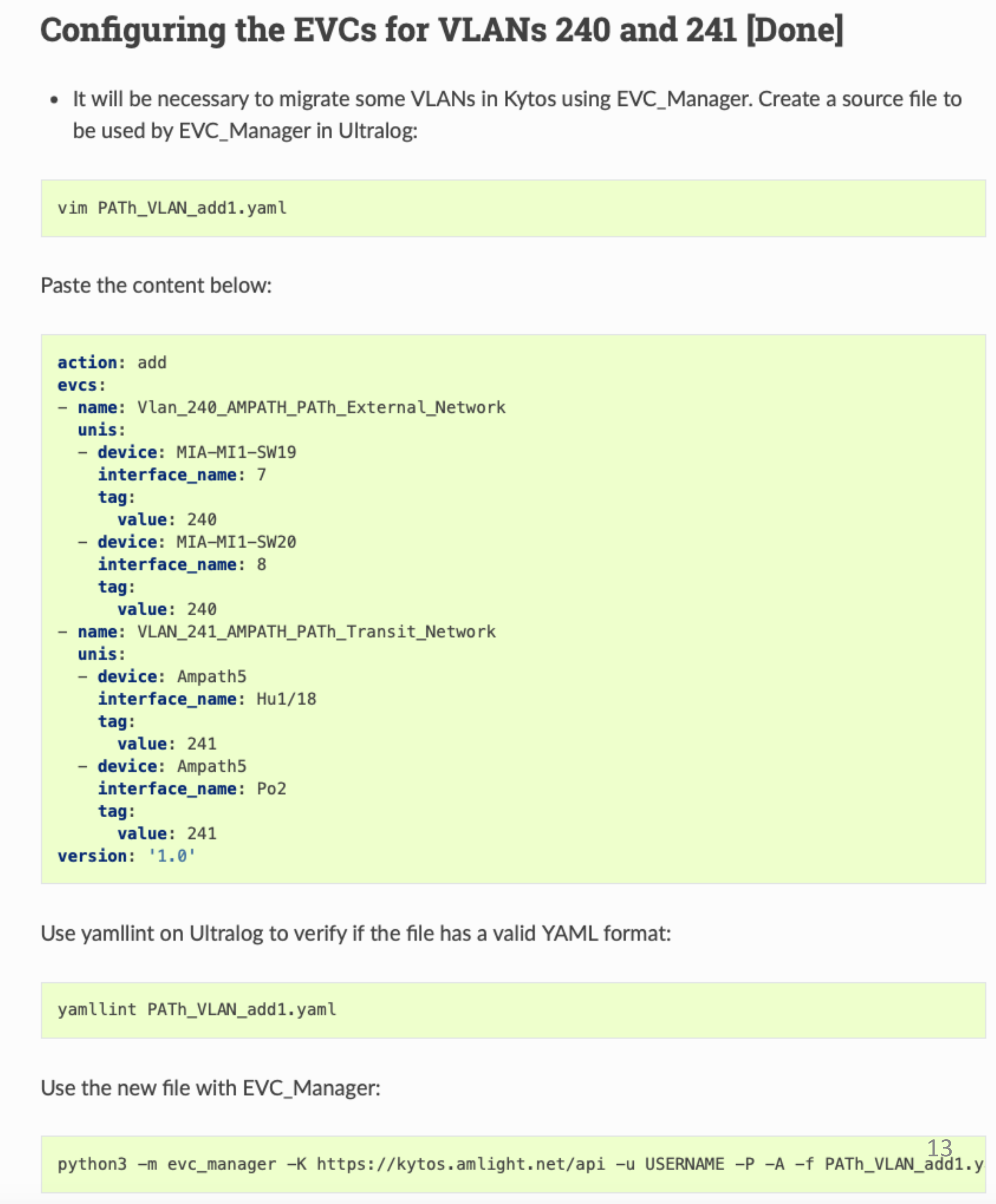
Creates a new circuit based on the endpoints and constraints given.

name required	string
frequency	string
uni_a › required	object (Endpoint)
uni_z › required	object (Endpoint)

```
{
  "name": "string",
  "frequency": "string",
- "uni_a": {
    "interface_id": "stringstringstringstrings
```

Kytos-ng user interfaces

- Example using evc-manager
  - Infrastructure as a code fashion
  - Systematic way to perform MW
  - Easy to document, review and execute



Configuring the EVCs for VLANs 240 and 241 [Done]

- It will be necessary to migrate some VLANs in Kytos using EVC_Manager. Create a source file to be used by EVC_Manager in Ultralog:

```
vim PATh_VLAN_add1.yaml
```

Paste the content below:

```
action: add
evcs:
- name: Vlan_240_AMPATH_PATh_External_Network
  unis:
  - device: MIA-MI1-SW19
    interface_name: 7
    tag:
      value: 240
  - device: MIA-MI1-SW20
    interface_name: 8
    tag:
      value: 240
- name: VLAN_241_AMPATH_PATh_Transit_Network
  unis:
  - device: Ampath5
    interface_name: Hu1/18
    tag:
      value: 241
  - device: Ampath5
    interface_name: Po2
    tag:
      value: 241
version: '1.0'
```

Use yamllint on Ultralog to verify if the file has a valid YAML format:

```
yamllint PATh_VLAN_add1.yaml
```

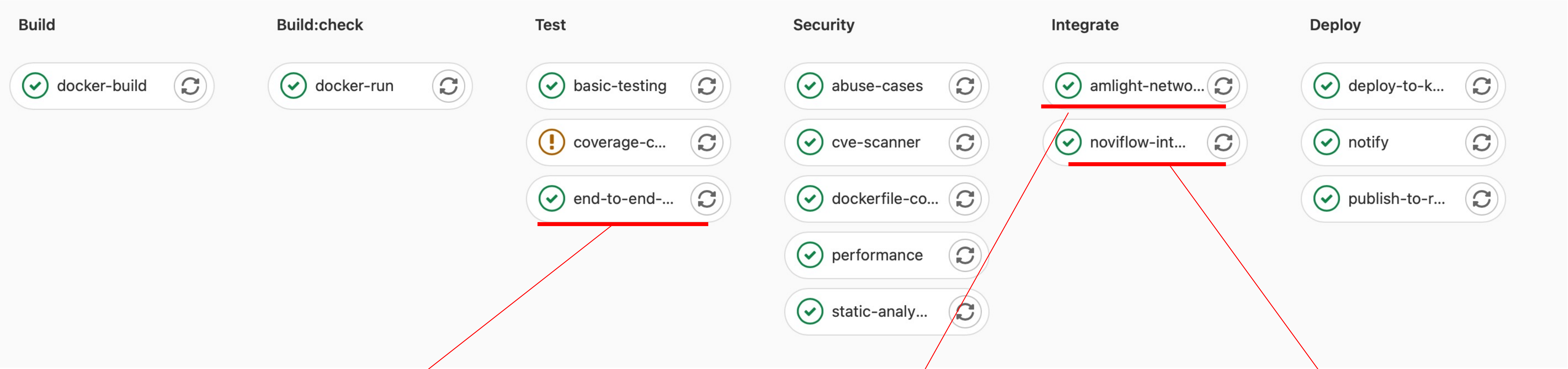
Use the new file with EVC_Manager:

```
python3 -m evc_manager -K https://kytos.amlight.net/api -u USERNAME -P -A -f PATh_VLAN_add1.y
```

How to keep kytos updated?

- Assuming Kytos-ng plays a critical role on AmLight network operations, how do we continuously update/deploy the SDN Orchestrator, providing an agile service delivery model for the users and operators in a integrated, secure, reliable and seamless manner?

Continuous integration pipeline



Build	Build:check	Test	Security	Integrate	Deploy
✓ docker-build	✓ docker-run	✓ basic-testing	✓ abuse-cases	✓ amlight-netwo...	✓ deploy-to-k...
		! coverage-c...	✓ cve-scanner	✓ noviflow-int...	✓ notify
		✓ end-to-end-...	✓ dockerfile-co...		✓ publish-to-r...
			✓ performance		
			✓ static-analy...		

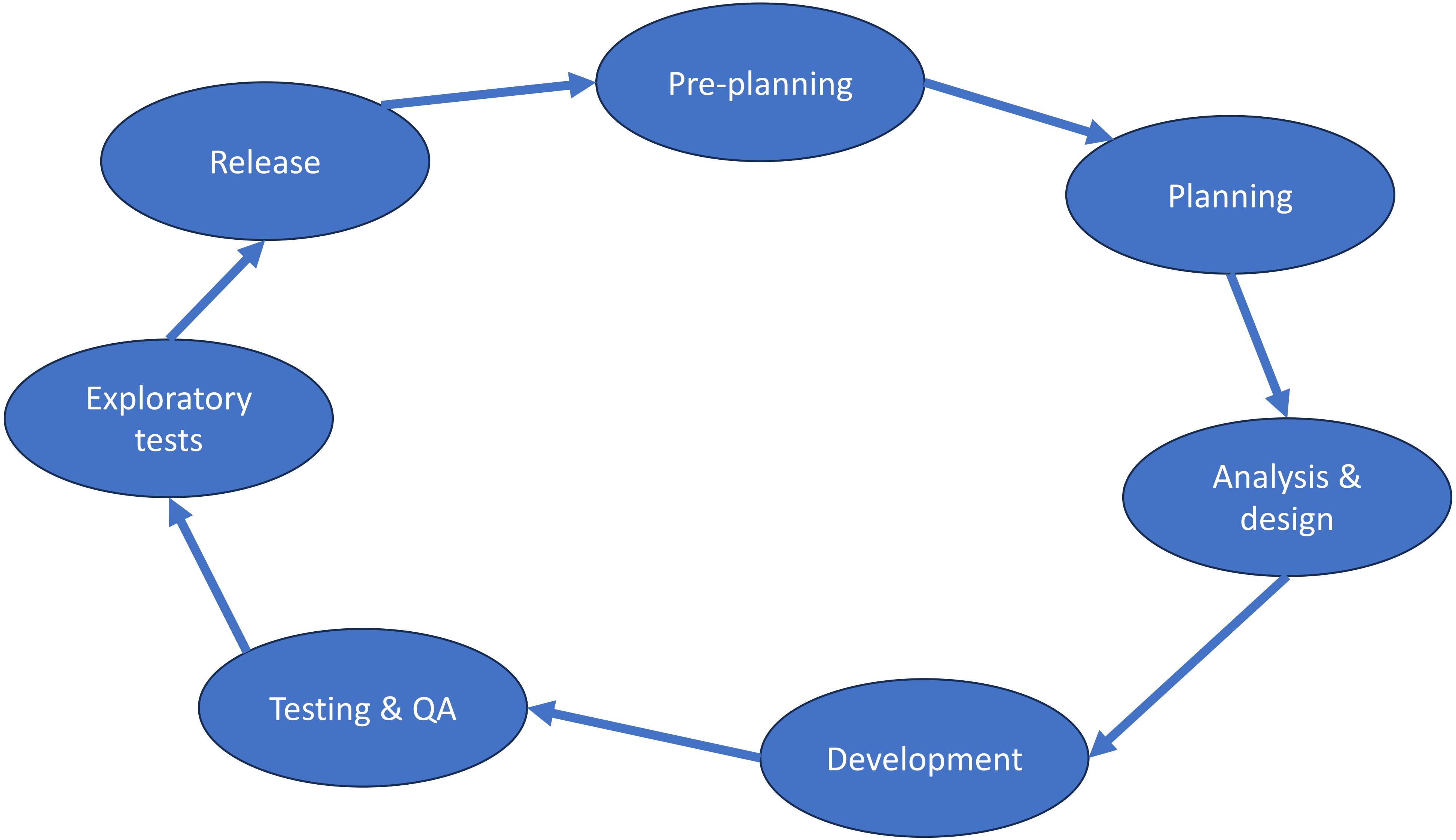
End-to-end tests: validate if the main services are working properly (Network emulator)

Virtual scenario/emulator using the same NOS in production, same topology, same user services

Hardware test: integration test with the same hardware (pre-production)

```
219 passed, 6 skipped, 11 xfailed, 5 xpassed in 3:10:22
```

Development model



Getting in touch

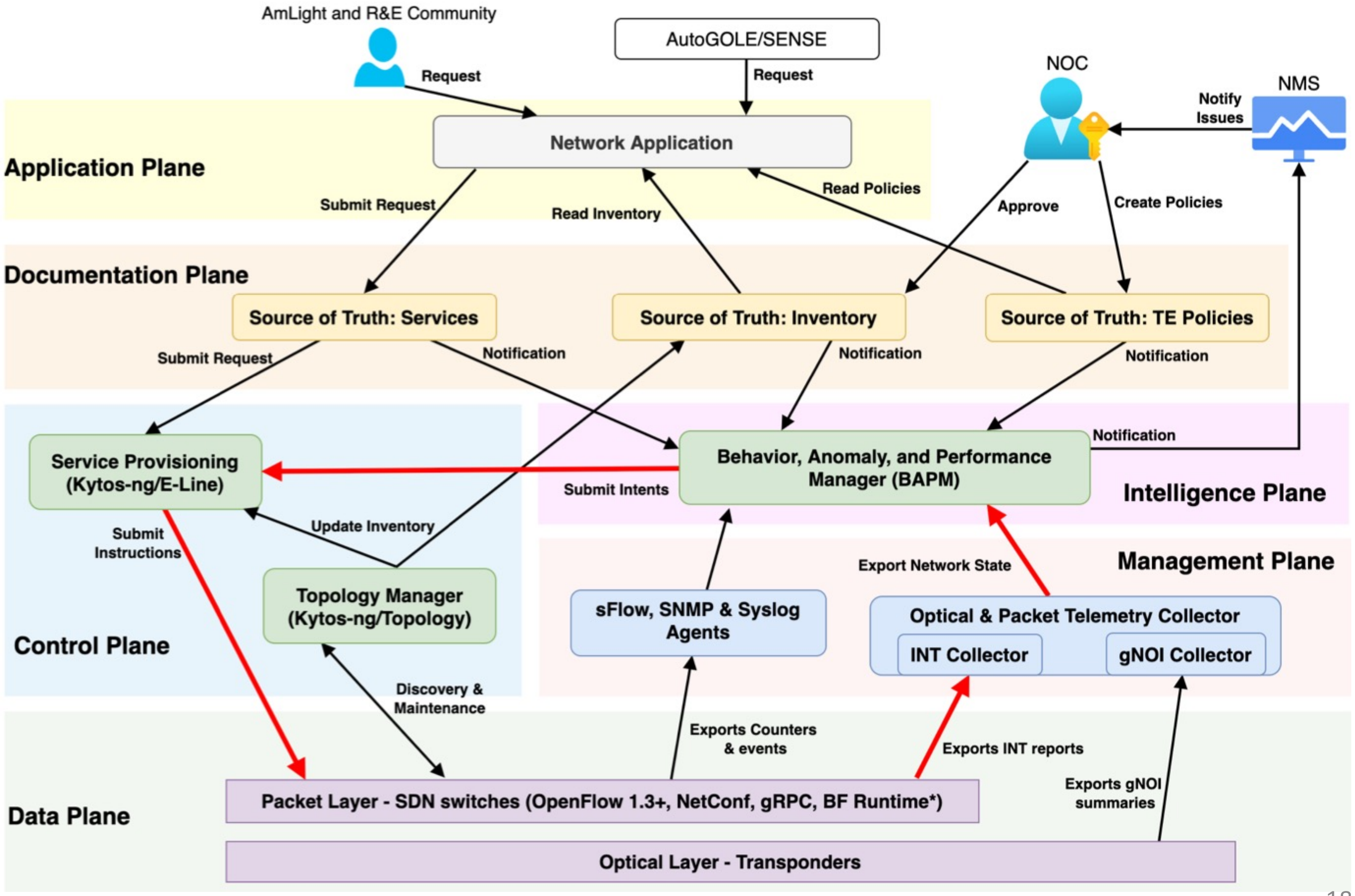
- Weekly meetings to discuss progress, issues, raise issues priority, reports from production environments
  - Every Monday, 2pm ET, 15hs BRT

- Slack channel: http://kytos-ng.slack.com

- Issues on Github: https://github.com/kytos-ng/

Evolving the AmLight-ExP SDN framework

- Evolving the SDN framework with six SDN planes to autonomically regulate AmLight-ExP network:
- Data Plane:
  - Exports counters from the Optical and Packet layers to the Management Plane
- Control Plane (CP):
  - Topology discovery and maintenance (Topology Manager)
  - Service Provisioning (submits instructions to Data Plane)
- Management Plane:
  - Exports network state to the Intelligence Plane:
    - Sampling counters; Optical and Packet telemetry
- Intelligence Plane:
  - Correlates events with inventory and traffic engineering policies from the Documentation Plane to learn the network state
  - Creates a closed-loop control for self-optimization
  - Submits requests to the CP if non-compliance
- The first Autonomic Function planned is to support L2VPNs fully managed by this architecture



2023 Roadmap for AmLight SDN network

- Deploy new switches and restructure the network in Jacksonville-FL, Atlanta-GA, Santiago-CL
- Deploy Kytos-ng release 2023.1
- Release 2023.2 of Kytos-ng SDN Controller
  - Enhances support for In-band Network Telemetry (INT), Bidirectional Forwarding Detection (BFD), and for VLAN ranges for point-to-point EVCs
- Assess the complexity of adding Barefoot Runtime (BFRuntime) as a southbound interface for provisioning
  - Objective is to evaluate the effort needed to support P4 natively and decommission OpenFlow in the near future

Other use cases

- Kytos-ng being used in Cybersecurity and Networking classes
  - Computer Networking at IC/UFBA (MATA59 2022.2) with Professor Leobino Sampaio and Italo Valcy
  - Regular projects at FIU Computer Science classes (SCIS Senior/Capstone Project) with professor Masoud Sadjadi
- Integration with AtlanticWave SDX 2.0 (first OXPO integrated)
- Proposal for “Hackers do Bem” RNP call (under review)
- Kytos being used to orchestrate non-programmable devices
  - Juniper devices, RARE/FreeRouter, Remessa 2.0 devices (!)

Conclusions

- With the new SDN architecture, AmLight expects to effectively load balance network services across links, while respecting user constraints and requirements
- Kytos-ng has proven to be very robust, reliable and flexible, while delivering services with great performance and protection
  - Development team working close to network engineers facilitate troubleshooting and speed up feature delivery
- Moving towards closed loop control reduces some time-consuming operational activities, which enables engineering team to focus on engineering and new services instead of manual activities

Thank you!

2023 WTR WORKSHOP TECNOLOGIAS DE REDE PoPBA

Kytos-ng at AmLight: the new generation of SDN orchestrator

Italo Valcy, Senior Network Engineer

italo@amlight.net