Using Kytos Platform to enable SDN Debugging
Raphael Cobe, Beraldo Leal, Carlos Eduardo Santos, Diego Rabatone Oliveira and Rogerio Iope
São Paulo State University - Center for Scientific Computing (São Paulo, Brazil)

SDN Debugging problem
Diagnosing a problem in an Software Defined Network (SDN) is still a challenging task. Most of the network administrators are stuck with tools such as ping, traceroute, SNMP or tcptrace to examine network traffic, checking connection and device status. In this poster, we introduce the Kytos SDN Platform and discuss a set of features that are currently being developed with it to ease the deployment of a SDN. The Kytos SDN debugging flavour – to be released in the first half of 2017. We are specially concerned with dealing with the following issues:

1) Race Conditions: this problem is commonly faced if controller applications expect some synchronization when installing rules in a Switch, as seen in (2). E.g., the application may not be ready to deal with a packet because a rule sent earlier has not been installed in the switch yet.

2) Controller Software Bug: a bug in the controller software may cause several network faulty behaviors, such as loops or routing problems. According to Zhi and Wei [3], this is the most common problem found on SDNs and the kind of test designed to discovery it is called White box [1].

Software-related defects are not found exclusively on the network administrator application side. In this new paradigm, faults can also be introduced by misbehaving software inside the network equipment, being the most common [1] problem with Equipment protocol implementation, as manufacturers are constantly including features and reimplementing parts of the protocol in their boxes. Such an active development cycle is very error prone.

Kyotos SDN platform
The Kytos SDN Platform provides a basic infrastructure for the development of modular SDN controllers and tools, using a loosely coupled asynchronous communication architecture for its components.

![Diagram of Kytos SDN Platform](image)

Kyotos separates the functionalities of the OpenFlow protocol from the controller core. Thus, we can enhance Kytos functionalities by adding new components responsible for providing specific protocol implementation. These components are called Network Applications – NApps. The communication between Kytos NApps and controller is done by shared message buffers. Also, NApps are allowed to register Representational State Transfer (REST) public endpoints, a feature specially useful to expose services outside the Kytos platform. We chose to use the Kytos SDN Platform as the base of our SDN debugging tool for its modular, yet simple architecture and detailed documentation.

References:

Acknowledgments
Thank you to all the people who contributed to this work.

Website: https://kytos.io/ IRC: #kytos @ freenode Mailing list: https://lists.kytos.io/