An Optical Network Emulator for Testing OpenROADM Controller and Training Students

Tianliang Zhang[1,*], Muhammad Ridwanur Rahim[1], Salem Blue Davidson[1], Brandon Hunter Grona-Gardom[1], Cristina Marie Kobierowski[1], Nathan Ellsworth[1], Gilles Thouenon[2], Christophe Betoule[2], Olivier Renais[2], and Andrea Fumagalli[1]

[1] *Open Networking Advanced Research (OpNeAR) Lab, UT Dallas, TX, USA*
[2] *Orange Labs, 2 Avenue Pierre Marzin, Lannion, France*

[*] *tianliang.zhang@utdallas.edu*

Abstract: The Optical Network Emulation (ONE) engine is a real-time and realistic platform for testing functionalities and FCAPS features of OpenROADM controllers. Its decentralized architecture, potential applications, and educational purposes are discussed in this paper.

1. Overview

Network emulation platforms like Mininet [1] have changed the way scientists explore and test software defined networking (SDN) solutions. With a simple software download one can instantiate a number of virtual hosts and switches to form a fully functional packet-switched network and apply a SDN controller (e.g., Ryu, OpenDayLight) to configure flow rules on said switches. On the other hand, operating with optical transport networks still requires access to a minimal set of specific internal parameters of actual optical devices — transponder and reconfigurable optical add-drop multiplexer (ROADM) to mention a few. Mininet-Optical [2] is a rare example of an SDN-based packet-optical emulator that supports integration with ONOS controller. The emulator is implemented using Mininet which enables packet exchange between Open vSwitches that would take place over the emulated optical circuits. A *single simulation engine* is applied to determine steady-state optical output signals and provide performance metrics like optical signal-to-noise ratio (OSNR), generalized OSNR (g-OSNR), and bit error rate (BER).

This paper presents the Optical Network Emulator (ONE) engine (Fig. 1(a)) which is designed to offer a low-barrier entry into software defined optical transport networking to the broader research and education community. The ONE engine realistically captures in real-time the behavior of the key optical devices that constitute a flex-grid wavelength division multiplexing (WDM) network and estimates their effect on the optical signal integrity as if it were transmitted in the real network. This goal is achieved by using a *decentralized implementation* of the optical devices behavioural models and their respective control firmware. The implementation of multiple software modules that run concurrently as containers — and implement basic functionalities that model the behavior of a multitude of optical network elements generally referred to as *E-Hardware* — offers the following unique advantages: a) cause-effect relationships between devices are closely captured following the same dependencies that are found in the real network, b) independent modules can be deployed by different researchers, each one providing their own specific expertise to the open-source development of the ONE engine, and c) the architecture is designed to scale and emulate hundreds of devices at once.

The role of the decentralized architecture of ONE is discussed for two distinct use cases, i.e., i) the use of ONE to test the correct implementation of functionalities and FCAPS features of the open-source Transport PCE (T-PCE) controller [3] and ii) the use of ONE to offer students the opportunity to be fully immersed in an emulated environment in which signal spectra are visualized using an emulated optical spectrum analyzer (E-OSA). The ambitious goal of concurrently fulfilling these two R&D and educational missions with a single tool requires an open-source approach that encourages the technical contribution from a number of interested researchers and educators over time.

2. ONE Decentralized Architecture and OpenROADM Use Case

The ONE engine decentralized architecture comprises a number of modules as shown in Fig. 1(a). These modules (generally referred to as E-Hardware) can be active (E-Transponder, E-ROADM, E-Amplifier, E-Regenerator, E-FiberSwitch, etc.) and passive (E-Fibers, E-Couplers, E-Attenuators, etc.). APIs are provided to add, remove, configure, and operate E-Hardware during run time, the same way one would do with actual optical hardware in the lab. In addition to these modules, the ONE engine makes use of interfaces to operate with other external

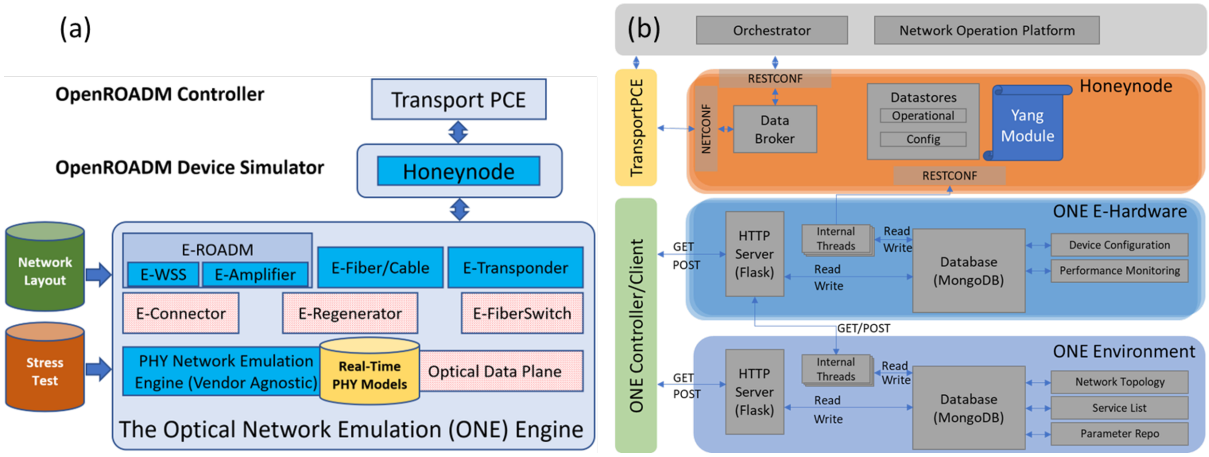

Fig. 1. (a) ONE decentralized architecture and key modules – dashed red modules not yet available and (b) OpenROADM network emulation through Honeynode and ONE engine

modules like Honeynode (firmware simulation of OpenROADM multi-vendor equipment), which in turn can operate with other modules like the OpenROADM compliant T-PCE controller. In turn, upper layer applications like the open-source NOP [4] collect real-time data, e.g., performance monitoring (PM) parameters through the northbound interface of T-PCE to create a data lake which is a centralized repository designed to store, process, and secure large amounts of structured and unstructured data concerning state information of the optical network. These platforms (T-PCE, NOP, and others) can be extensively tested using the ONE engine to account for various network configurations including large systems with hundreds of network elements and active wave services. Alternatively, these platforms can only be tested on relatively small lab testbeds and early field deployments.

Fig. 1(b) shows the key software modules and the integration of Honeynode and ONE engine to work as a whole. The ONE engine is designed to be vendor and standard agnostic and can be therefore adapted to comply with the OpenROADM Honeynode module. Data structures are defined in YANG models for ease of management and scaling. the ONE Environment and ONE E-Hardware software modules run as Docker containers with Flask as a HTTP server implemented to provide the REST API for external access to configure or read data in the provided database (MongoDB).

The ONE Environment is a key module that ensures cause-effect between E-Hardware components and the optical signals as they propagate through the optical network[1]. Signal spectra are represented by polynomial terms as described in Section 3, which facilitate the propagation of each signal between E-Hardware components till it is delivered to the destination port. Any interaction with source and destination ports is through REST API calls to the corresponding E-Hardware. The ONE Environment receives external requests from the ONE controller or any REST API call client, such as adding, removing, cutting or injecting external noise into a fiber.

The ONE E-Hardware has a structure similar to that of the ONE Environment while its functionality varies from device to device. Each E-Hardware software module accounts for the main physical limitations as they are found in the corresponding actual hardware equivalent. In simple words, the E-Hardware module captures the hardware behavior through analytical and numerical models while the corresponding Honeynode module simulates the software and firmware (the 'binaries') that run on the hardware. Each E-Hardware is characterized by its own PMs. For example, the E-Transponder assigns an internal thread to each port in order to examine the receiving signal quality and compute performance characteristics like OSNR and pre-FEC BER.The E-ROADM assigns an internal thread to each port to examine the receiving signal(s) and route it through internal wavelength selective switches (WSS) to the destination port. Signal spectra are modified by applying ASE noise, insertion loss, and WSS filtering effect [5] (see Section 3 for an example on WSS filtering).

The Honeynode device simulator implements the YANG models of OpenROADM devices — including transponders and ROADMs — which are accessible by T-PCE controller [3] through NETCONF APIs. As shown in Fig. 1(b) the ONE engine is integrated with FD.io HC2VPP-based Honeynode [6] running as Docker container. The integration of Honeynode and ONE provides a powerful software-only platform that includes handling of real-time status and PMs for thorough testing of service assurance capabilities in T-PCE. Upper layer applications can use their external APIs to perform various functionalities, such as network monitoring.

[1]Network Topology stores the link source, destination, length and fiber type. The fiber characteristics are saved in the Parameter Repo.

3. Signal Spectrum Modeling with Polynomial Coefficients

A signal power spectrum and its changes (e.g., WSS filtering) are modelled using polynomial terms in ONE. The benefits include: a) it provides flexibility in modelling all shapes of the signal, b) interaction with optical components in the network can be calculated in power-frequency domain, and c) the shape of the signal can be visualized anywhere in the network, which is analogous to putting a probe in the network using an emulated OSA (to be implemented). By collecting data samples from real transponders, different types of signals can be fitted by polynomial coefficients and stored in the repository. Using polynomial coefficients with an optimal order to fit the data samples gives us any desired resolution of the signal frequency representation. In Fig. 3(a, c), two 100G DP-QPSK signals with different initial power levels generated from a non-commercial transponder and sampled by OSA are fitted by 50 orders of polynomial coefficients. Polynomial representations have been found to fit well on different signal spectra. Similarly, we use polynomial coefficients to represent the WSS filter against its analytical model [5], but could also be used to fit experimental data. The polynomial coefficients of a signal passing through a WSS filter can be calculated through the convolution of two sets of coefficients: one from the input signal, and the other from the WSS. The filtering effect on the 100G signal through 2 to 12 WSS devices calculated by polynomial coefficients is shown in Fig. 3(a, c). The WSS filter is chosen to have a 6 dB bandwidth of 37.5 GHz and a roll-off factor of 10.4 GHz. Fig. 3(b, d) show the filtering effects when the signal spectrum is offset to the right from the central frequency by 5 GHz. Both plots show that the polynomial coefficient representation successfully models the signal shape change when the signal is filtered by the WSS filter. The polynomial coefficients of the product of two terms can be time efficiently computed using Fast Fourier Transform (FFT) as opposed to convolution when the number of polynomial coefficients is large, e.g., in the order of 1,000 or higher. Both methods take ∼1 ms for computing each product (up to 650 coefficients) on Intel Core i5-4210U CPU@1.70GHz.

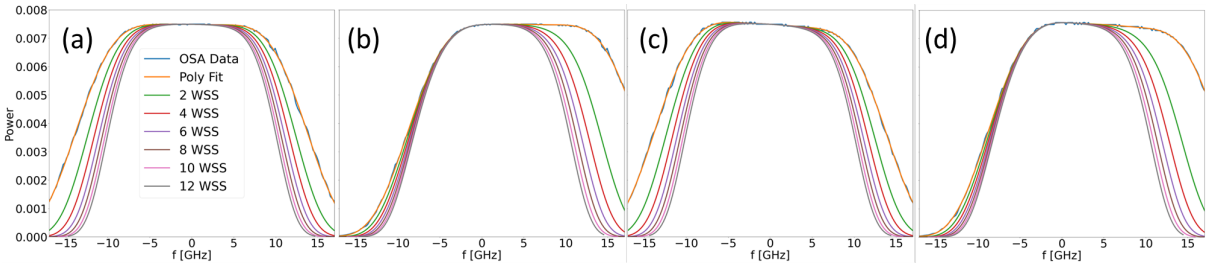

Fig. 2. (a,c): Normalized signal power spectra using polynomial terms for two launched power levels as they cross increasing number of WSS devices; (b,d): Signal central frequency offset is 5 GHz

4. Summary

The ONE decentralized architecture enables software modules to be integrated and achieve emulation of complex WDM networks. The integration of E-Hardware and Honeynode provides a testing environment for validating advanced functionalities in the OpenROADM T-PCE controller and related upper applications. Real-time emulation of transponders and ROADMs is achieved using multiple containers and polynomial signal spectrum representation. Polynomial coefficients provide an efficient way to represent the signal state across the various network sections in real-time, i.e., coefficients are passed from device to device to capture cause-effect dynamics, and polynomial representation offers any desire frequency resolution to inspect signal using emulated probes.

Acknowledgments: This work is supported in part by NSF grant CNS-1956357 titled “Optics without Borders.”

References

1. Mininet, “Mininet,” http://mininet.org/ (2022). Last retrieved 10/15/2022.
2. B. Lantz *et al.*, “Demonstration of software-defined packet-optical network emulation with mininet-optical and onos,” in *2020 Optical Fiber Communications Conference and Exhibition (OFC),* (2020), pp. 1–3.
3. Transport PCE, “Transport PCE,” https://docs.opendaylight.org/projects/transportpce/en/latest/user-guide.html (2021). Last retrieved 10/18/2022.
4. N. Ellsworth and et al., “Using network operations platform and orchestrator to enhance programmable openroadm optical networks,” in *2022 Optical Fiber Communications Conference and Exhibition (OFC),* (2022), pp. 1–3.
5. C. Pulikkaseril *et al.*, “Spectral modeling of channel band shapes in wavelength selective switches,” Opt. Express **19**, 8458–8470 (2011).
6. Honeynode, “Honeynode,” https://gitlab.com/Orange-OpenSource/lfn/odl/honeynode-simulator (2021). Last retrieved 10/18/2022.