Enhancing Optical Network Emulation (ONE) Engine with Multi-Container Scalability and B400G Signal Modeling

Muhammad Ridwanur Rahim, Aparaajitha Gomathinayakam Latha, Tianliang Zhang, Marco Tacca, and Andrea Fumagalli
The University of Texas at Dallas, Richardson, TX, USA
{*muhammadridwanur.rahim, aparaajitha.gomathinayakamlatha, tianliang.zhang, mtacca, andreaf*}
@utdallas.edu

ABSTRACT
The Optical Network Emulator (ONE) engine is a multi-container software-only solution, designed by the OpNeAR lab at UT Dallas to perform various functionalities of a wavelength division multiplexing (WDM) network in real-time. This paper presents the first real-time demonstration of the ONE engine, customized to emulate a replica of the collaborative WDM network testbed jointly presented by the OpenROADM MSA and IOWN GF teams at OFC'24. Leveraging newly developed modeling techniques the software accurately replicates the behavior of transponders, Reconfigurable Add Drop Multiplexing (ROADM) devices, and optical amplifiers. Additionally, the paper explores the scalability of the ONE engine implementation, demonstrating its potential to support larger and more complex network configurations using a single server system.

By enabling practical hands-on experimentation with software-defined networking (SDN) technologies applied to WDM networks, the ONE engine has significant implications for education and industry.
Keywords: Optical Network Emulation, Real-time Emulation, Multi-container, WDM Networks, B400G Transmission

1. INTRODUCTION

The ONE engine comprises multiple software modules that run concurrently as containers and implement basic functionalities that model and reflect the behavior of a multitude of optical transport network elements. These modules, generally termed E-Hardware, include active and passive components. In the ONE engine, the spectral characterization of these components is modeled through polynomial fitting [1] obtained through experimental data from physical devices.

Unlike other software emulators such as Mininet-Optical [2], the ONE engine relies on multiple docker containers. As a result, the ONE engine enables dynamic device reconfigurability during runtime, made possible by the decentralized nature of its implementation, wherein each E-Hardware is defined using an individual container. This trait is analogous to manual interactions with physical components. For example, engineers and technicians can simultaneously log into multiple devices and make real-time adjustments, exactly how they would operate in a real transport network. Hence researchers and students alike can identify and address issues or establish new transport services without shutting down the network emulation. This interactive learning environment facilitates a more realistic understanding of the components and functionalities of a typical optical WDM transport network.

The ONE engine can generate signal spectra for any given parameter, imitating real-life components. Users can observe the power spectral density (PSD) of any composite signal as it traverses the optical transport network, along with any change made to add or modify existing transport services in real-time. This paper presents a first practical use case by emulating a WDM network replicating the collaborative testbed demonstration by the OpenROADM MSA and IOWN GF engineers at OFC'24. The ability of the software implementation to emulate a larger number of optical devices running on a single server is then discussed.

2. THE OPTICAL NETWORK EMULATOR (ONE) ENGINE

As previously mentioned, the main objective of the ONE engine software is to emulate the behavior of optical hardware components as individual and independent containers while at the same time accounting for their respective physical interactions when they are integrated to form a single optical transport network.

2.1 ONE Engine Key Modules

The following hardware components are accounted for by the *distributed container-based architecture* of the ONE engine.

Active hardware components include transponder, ROADM, in-line amplifier, and variable optical attenuator (VOA). Each component is individually emulated by an E-Hardware software module. Each E-Hardware software module accounts for the main functionalities as it is found in the corresponding actual hardware equivalent, e.g., generating an optical signal with a specific baud rate, transmit power and frequency, routing signals through an E-ROADM node that incorporates wavelength selective switch (WSS) filtering and pre/boost-amplifiers, etc.

979-8-3503-7732-3/24/$31.00

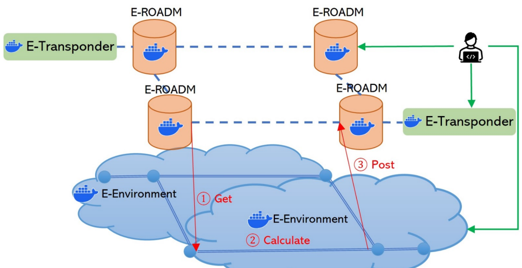

Figure 1: ONE engine software architecture

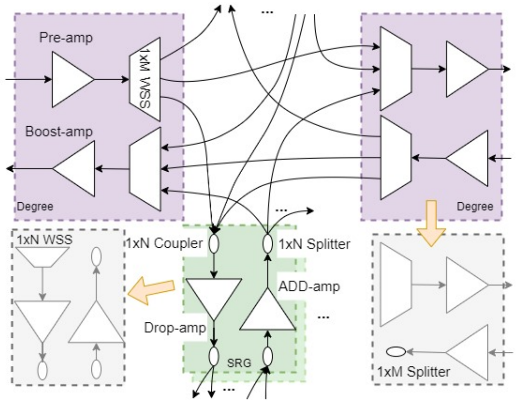

Figure 2: ROADM architectures and amplifier types: ROADMs may have different implementations on degree and shared-risk-group (SRG) depending on vendors' choices, shown in grey

To ensure real-time execution, these models must adopt a certain level of abstraction that does not require the emulation of individual transmitted bits, but rather, a PSD modeled using a polynomial fitting technique while also focusing on the statistical characterization of the hardware behavior. Any external changes by the user will be recorded in the device configuration and applied by the auto-update procedure defined for the E-Hardware module as described in [3].

Passive hardware components include fiber connector, attenuator, and splitter. These E-Hardware modules account for the main effects of these components on signal integrity, such as power loss and non-linear interference (NLI). Parameters defined in the E-Hardware module may be changed over time during the experiment to reflect the outcome of applied external actions, for example, mechanical stress applied to a fiber cable that would change the characteristics of the medium.

The ONE Environment component is the central module that ensures coordination among E-Hardware components and accounts for the cascaded effects caused by these components on the optical signals as they propagate through the emulated optical network. The E-Environment component constantly communicates with each E-Hardware component to provide and obtain critical parameters concerning propagating optical signals, which are used to model the behavior of the optical network as a whole. The E-Environment component is also responsible for storing and instantiating the network topology, which lists the E-Hardware containers and accounts for their interconnections in the network.

Once launched, the E-Environment component periodically checks all the links in the emulated network using an auto-update procedure for every fiber link present in the emulated network. Running in multi-threading mode, it will retrieve the output signal(s) from one link's source (Step 1: the `Get` in Figure 1), apply the physical impairment in the fiber to the signal, for example, loss (Step 2: the `Calculate` in Figure 1), and post the signal to the destination (Step 3: the `Post` in Figure 1). Getting and posting signal parameters are executed by calling the RestAPIs on E-Transponders and E-ROADMs.

2.2 ONE Engine Data

This section focuses on two main E-Hardware components – the E-Transponder and the E-ROADM – and details how they reproduce data similar to actual components. This process is facilitated by using a NoSQL database, i.e., MongoDB.

Since an E-Transponder must replicate any user-defined signal spectra, mirroring the capabilities of a real optical transmitter module, it is equipped with multiple sets of polynomial coefficients, allowing signal spectra generation for any given configuration. These coefficients are stored in the ONE engine memory upon launching, enabling users to update the signal during run time (based on the capabilities of the transponder) without requiring further access to MongoDB.

Similarly, in the case of an E-ROADM, the emulated device contains the polynomial coefficients of multiple reconfigurable WSS transfer functions. With the current implementation, the following options are available: a 100 Gb/s signal can pass through a 37.5 GHz WSS, a 400 Gb/s signal can pass through an 87.5 GHz WSS, and a beyond 400 Gb/s signal can pass through a 150 GHz WSS, subject to the modulation format and channel spacing assigned.

The ONE engine MongoDB can be populated using either generalized hardware models based on theoretical and hypothetical results or specific hardware models derived from experimental and commercial devices. Results reported in this paper are obtained using a combination of these options.

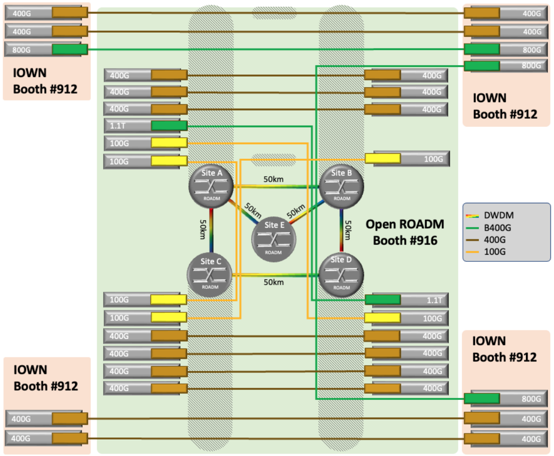

Figure 3: Topology replica of the OFC'24 demo jointly staged by OpenROADM MSA & IOWN GF as implemented in ONE Engine

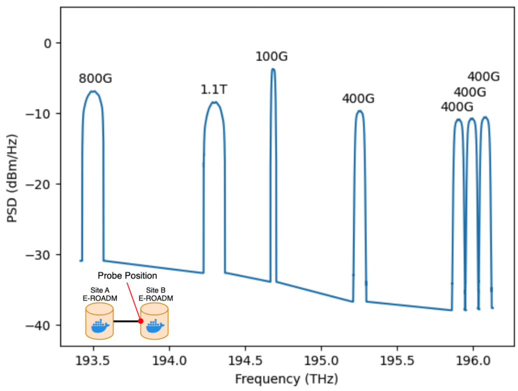

Figure 4: ONE-generated composite signal on Site-A to Site-B span in Figure 3

3. PHYSICAL COMPONENT MODELING

Transmitter. As previously mentioned, each transmitted signal is modeled through polynomial fitting [1] of its PSD. In its simplest and current implementation, ONE utilizes a single set of coefficients to accurately represent the shape of the PSD coupled with a scalar to report the total power in the -35 dBm to +10 dBm range. The dependency of the transmitter noise level on the signal launched power is modeled using polynomial fitting of degree 10. A few normalization techniques, the details of which are beyond the scope of this paper, facilitate adjustments for varying power levels, allowing accommodations for amplifiers, insertion losses, and attenuation. Polynomial coefficients representing the signal PSD are modified when passing through a WSS according to the WSS transfer function applied [3].

Wavelength Selective Switches. An analytical model of wavelength selective switch (WSS) [4] is fitted with polynomial coefficients of degree 50 to accurately capture their transfer function[1]. The ONE engine currently incorporates models of multiple WSS filters, each with a different 6 dB bandwidth, ranging from 37.5-150 GHz, and a roll-off factor of 10.4 GHz. Polynomial fitting is based on a linear scale, with data normalized to the maximum value of the data set, and alignment at the zero-frequency point.

Amplifiers. Figure 2 showcases the use of four types of amplifiers inside a ROADM device: Pre-amp, Boost-amp, Add-amp, and Drop-amp. Presently, the ONE engine models these amplifier types based on experimental data. The OpenROADM MSA defines the C-band from 191.325-196.125 THz, whereas, our experimental data for amplifier gain and noise figure ranges from 192.1-196 THz. Thus extrapolation of available experimental data was required to cover the OpenROADM-defined C-band range. More specifically, a 2-D polynomial fitting approach is applied to model the actual gain (and noise) at any given frequency and for any desired set gain applied to the amplifier. The details of the extrapolation step and 2-D polynomial fitting technique are beyond the scope of this paper.

4. TWO USE CASES

The ONE engine is designed for scalability. In this study, the emulation of two networks is reported. One emulated network reproduces the testbed jointly demonstrated by the OpenROADM MSA and IOWN GF groups at OFC'24. The other emulated network reproduces a line network (point-to-point) with several transmitted signals filling up the entirety of the C band, emphasizing the scalability potential of the ONE engine design.

4.1 Emulating Optical Transport Network Testbed Showcased at OFC'24

At the 2024 Optical Fiber Communications Conference, in collaboration with the IOWN GF group, the Open-ROADM team presented a WDM network testbed showcasing multi-vendor interoperability at data rates ranging from 100 Gb/s to 1.2 Tb/s, along with several other functionalities (e.g., IPoDWDM and enhanced network operations platform). Leveraging this opportunity, the ONE engine is configured to recreate the same physical components and topology layout chosen for the OFC'24 testbed. The set of wavelength services emulated, their assigned frequencies, baud rates, and physical paths are also chosen to match those used in the OFC'24 testbed.

[1]To represent more realistic scenarios, WSS devices can alternatively be characterized using experimental data when creating the polynomial coefficients.

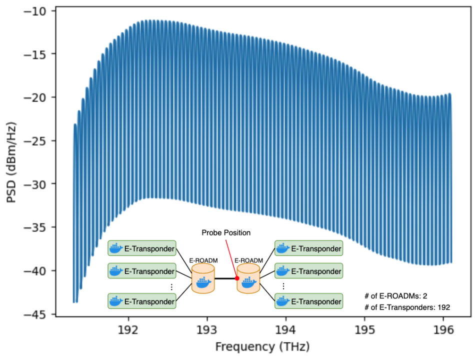

Figure 5: Line-network with full utilization of C band using 96 E-Transponder pairs at 100 Gb/s

TABLE I: CPU and Memory Usage

Use Case	E-Hardware	∼CPU usage	∼RAM
OFC'24	Transponder	1% /core	100 MiB
	ROADM	100% /core	150 MiB
	Environment	41% /core	155 MiB
	Total	10%	5.3/31.3 G
Line-network	Transponder	1% /core	100 MiB
	ROADM	160% /core	118 MiB
	Environment	160% /core	310 MiB
	Total	40%	21/31.3 G

Figure 3 shows the network topology emulated by the ONE engine, consisting of five ROADM nodes. Each ROADM node in the figure is assigned a shaded grey area. A wave service crossing a specific grey area indicates that such service is routed through the add/drop port (SRG) of the corresponding ROADM node. Figure 4 shows the emulated composite signal on the span connecting Site-A to Site-B. The signals exhibit varying power levels since we intentionally did not apply channel power equalization to visually highlight the effect of non-uniform amplifier gain in the emulation.

4.2 Scalability Test

In this section, the ONE engine is applied to emulate a line network comprising 96 E-Transponder pairs, with every pair directly connected using 100G/s signals routed through the WDM line network span. Figure 5 illustrates the composite signal received at the probe position placed at the degree input port of the right E-ROADM in the emulated line network. At the left E-ROADM, the 100G signals are launched by the 96 E-Transponders using the same power level. As shown in Figure 2, the launched signals must be amplified by two amplifiers (ADD-amp and Boost-amp) before reaching the fiber span heading to the right E-ROADM. As previously mentioned, channel power equalization is not applied in the emulation, and the resulting composite signal in Figure 5 displays the typical amplifier non-flat gain behavior across the C-band.

Table I shows the CPU and memory usage for each of the E-Hardware modules in the two use cases. In both cases, the E-Environment container uses considerable resources since it must perform the Auto-update procedure and computations to account for the signals propagating between E-Hardware components. The ONE engine can support at least 250 E-Hardware modules on a single server, thus enabling the emulation of even more complex networks. The E-ROADM container is also consuming considerable resources to account for the signals propagating through them and their spectral density changes due to the WSS transfer functions.

5. SUMMARY

This paper reports the first WDM network testbed emulated by the multi-container ONE engine. In addition, an emulated line network showcases the ONE engine's ability to emulate 200 network components using a single server. Composite signals are automatically generated by the ONE engine reproducing typical spectral patterns, e.g., amplifier's non-flat gain and signals' spectral allocation. These results illustrate the ONE engine's current features and versatility to become an instrument for quickly and inexpensively conducting experiments with emulated WDM networks. Future work will focus on adding channel power equalization in the ROADM node and modeling of Non-Linear Interference (NLI) noise in the optical fiber span.

ACKNOWLEDGMENTS

This work is supported in part by NSF grant CNS-1956357 titled "Optics without Borders."

REFERENCES

[1] Latha, Aparaajitha Gomathinayakam, et al. "Accurate Representation of Signal Power Spectral Density in the Optical Network Emulation (ONE) Engine." 2023 23rd International Conference on Transparent Optical Networks (ICTON). IEEE, 2023

[2] Lantz, Bob, et al. "Demonstration of software-defined packet-optical network emulation with mininet-optical and ONOS." Optical Fiber Communication Conference. Optica Publishing Group, 2020

[3] Zhang, Tianliang, et al. "An optical network emulator for testing OpenROADM controller and training students." 2023 Optical Fiber Communications Conference and Exhibition (OFC). IEEE, 2023

[4] Pulikkaseril, Cibby, et al. "Spectral modeling of channel band shapes in wavelength selective switches." Optics express 19.9 (2011): 8458-8470