OPEN ACCESS

University of BRISTOL

Peer reviewed version

Link to published version (if available):
10.1109/IMOC43827.2019.9317692

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

Enabling User-Friendly EON Simulations with a Graphical Interface for ElasticO++

Yrui Giovan Neris
Dept. of Electrical Engineering
Federal University of Espírito Santo
Vitória - Brazil
yrui.neris@aluno.ufes.br

Rodrigo Stange Tessinari
High Performance Networks Group
University of Bristol
Bristol - United Kingdom
rodrigo.tessinari@bristol.ac.uk

Anilton Salles Garcia
Dept. of Electrical Engineering
Federal University of Espírito Santo
Vitória - Brazil
anilton@inf.ufes.br

Marcelo Eduardo Vieira Segatto
Dept. of Electrical Engineering
Federal University of Espírito Santo
Vitória - Brazil
segatto@ele.ufes.br

Marcia Helena Moreira Paiva
Dept. of Electrical Engineering
Federal University of Espírito Santo
Vitória, Brazil
marcia.paiva@ufes.br

Abstract—With the advancement of technology and telecommunication networks, there is an increasing need to develop tools to assist in decision-making, and to enable new research developments. The ElasticO++ simulation tool makes easier the development of new research in optical telecommunications networks. In this work, we implemented a graphical interface that facilitates and speeds up the simulation process in ElasticO++, allowing to modify its parameters intuitively, to run simulations for a large set of networks simultaneously, and to visualize the results. To demonstrate how the implemented interface improves the usability of ElasticO++, we present an experiment with the NSFNET network together with a set of 70 topologies generated by adding a single link in all possible ways in the NSFNET network.

Index Terms—ElasticO++, Simulation tool, Telecommunications, Optical networks.

I. Introduction

The growth in Internet usage has increased the importance of telecommunication networks, and modified the data traffic characteristics, increasing the demand for information. In the coming years, the trend for traffic is to reach Exabytes per month in 2020 [1]. The constant and dynamic alteration in the traffic demands requires the implementation of traffic engineering techniques that can meet and follow the changes in the scenario of telecommunications networks.

Nowadays, optical networks meet traffic demands effectively. Most telecommunications networks use Wavelength-Division Multiplex (WDM), with transmission capacity per channel of up to 100 Gb/s [2]. With the advent of Elastic Optical Networks (EON), the 100 Gb/s channel transmission capacity of WDM networks increased to 1 Tb/s in EON networks, due to the flexibility of channel spacing [3].

With the emergence of elastic optical networks, new challenges have emerged, such as the Routing and Spectrum

Assignment (RSA) problem [2], which consists of finding sufficient amount of resources to meet the connection requests. The available resources are limited, and the technique used to route and allocate information in the spectrum along the way influences network performance, where the lack of spectrum or other constraints imposed by the technology can cause blocking of requests and spectrum fragmentation [4].

Network performance is affected by many diverse factors, such as changes in spectrum management techniques [5] and topology [6]. Therefore, it is necessary to use tools to simulate the behaviour of communication networks and to verify the impact of the changes on network performance.

The ElasticO++ [7] simulation tool enables to test a range of algorithms, parameters, and topologies for routing, modulation, spectrum assignment, and defragmentation, helping users in their research projects and being a starting point for new researchers. The tool enables the implementation and comparison of new algorithms with the results of algorithms previously implemented.

In this work, we implemented a graphical interface that facilitates and speeds up the simulation process in ElasticO++, allowing to modify its parameters intuitively, to run simulations for a large set of networks simultaneously, and to visualize the results as graphs. To demonstrate how the implemented interface improves the usability of ElasticO++, we present an experiment with the NSFNET network together with a set of 70 topologies generated by adding a single link in all possible ways in the NSFNET network.

II. EON Simulation Framework

ElasticO++ is an Elastic Optical Network Simulation Framework for OMNeT++ [8], and it aims to facilitate and speed up the work required to create and test new algorithms, reducing the time developing code for getting results. ElasticO++ is compatible with OMNeT++ 5.0. The tool can be used without the need of changing the source code directly,

and users only need to change the configuration files to get results. The tool has the following functionalities:

- Perform Elastic Optical Networks simulations
- Support fragmentation and defragmentation;
- Allow easy test of algorithms;
- Be flexible and easily upgraded if required;
- Capable of running batch simulations;
- Support of charts and statistics generation;

The tool can handle specific characteristics of elastic optical networks (routing, modulation, spectrum management, and spectrum allocation), and it enables interaction and alterations by researchers.

A. OMNet++

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. Its code is free and can be used under the Public Academic License which makes the software free for non-profit use [7].

To perform simulations in OMNeT++, in addition to the implementation of the modules, two main files are required: the Network Description (.ned) file containing the network description (topology), the simulated modules and the connections between them; and the initialization (.ini) file, that contains the necessary parameters for the simulation. Also, it is possible to consider additional configuration files.

Using ElasticO++ requires the user to have some knowledge about programming and about the tool, where it is necessary to change the network description and initialization files, which may limit the use of the tool. Given these characteristics, we have developed an interface capable of managing and executing the simulations, without the need to change the configuration files directly. This way, a wider audience of users is reached, and the benefits and goals of the ElasticO++ tool are enhanced.

B. The proposed interface

With the use of the interface, user interaction with OMNeT++ is no longer necessary. Users interact only with the interface, that communicates with ElasticO++, as shown in Fig. 1.

The interface allows the user to change the ElasticO++ parameters, to insert new topologies, to combine algorithms already implemented in the tool, and to run simulations. After defining parameters and topology, the interface communicates with ElasticO++, creates and updates the configuration files, and executes the necessary processes to run the simulations. At the end of execution, ElasticO++ returns the results, and the interface shows the results to the user in a visual form.

The interface was developed using the programming language Python 2.7.Fig. 2 shows the main parameters that can be changed in the interface:

- *Load settings:* loads existing configurations;
- *Add and select networks:* allows the inclusion of new topologies;
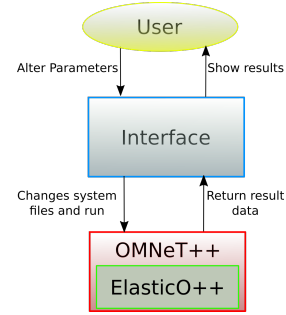

Fig. 1. User interaction with ElasticO ++ using the interface.

- *Algorithms:* allows choosing multiple allocation algorithms. An allocation algorithm is a combination of routing, modulation, spectrum management, and spectrum allocation, implemented in the tool;
- *Number of paths:* K value when using the K-Shortest Paths algorithm;
- *Number of repetitions:* each repetition uses a different random seed for each simulation;
- *Number of simulations:* total number of connection requests;
- *Number of resources in each fiber:* bandwidth available per fiber, in 12.5 GHz slots;
- *Network load:* network load range (in Erlang units);
- *Allowed services:* types of services (in Gbps);
- *Traffic pattern:* proportional generation ratio of each service type;
- *Charts:* types of charts generated automatically after the simulation.

Fig. 2. List of parameters that can be changed in the interface.

The developed interface adds flexibility and improves the overall usability of ElasticO++, especially when adding new algorithms and managing existing ones. Furthermore, it automates the whole simulation process, including, but not limited to, the creation of configuration files, and the execution of tasks required for simulating networks and demonstrating the results. Therefore, it facilitates and speeds up the simulation process.

III. EXPERIMENTS AND RESULTS

The objective of this Section presenting the features of the interface and how they enhance the ElasticO++ overall experi-

ence. For our tests, we compute and analyze the blocking rate results of two sets of topologies using an algorithm available at the tool.

For our experiments, we use the developed interface to test the usability of the tool and present the obtained results. All simulations were done on a Windows Server 2012 R2 64-bit using Intel (R) Xeon (R) CPU E5-2620 v4 @ 2.10GHz with 8 threads and 4 GB of RAM. With the use of multiple threads, it is possible to reduce execution time, each simulation is executed in a single thread, and for our case 8 simulations can be executed simultaneously (the total number of simulations varies according to the simulation parameters).

As a way of assessing the usability of the interface, we analyze the blocking rate results for a large set of networks. For that, a set of 70 topologies was generated from the addition of a single link in all possible ways in the NSFNET network, that has 14 nodes and 21 links. Then, we compare the blocking rate results of these 70 generated topologies of 14 nodes and 22 links with the results of the original NSFNET topology.

To import all topologies using the interface, it is necessary to configure input files with the format "Source Node, Destination Node," representing the links, and import the topologies by selecting the files, being possible to import all of them at once. In this case, the interface will create 71 network description files automatically, that is much faster than the manual way.

For the simulation, we configured 360 slots (360 $\times$ 12.5 GHz = 4.5 THz) per fiber and 200000 connection requests, and each simulation was repeated 5 times. The requests demand 4 types of traffic, being 40, 100, 400 and 1000 Gb/s, with a traffic pattern of 4:3:2:1 respectively. For the allocation and routing of the information in the spectrum, we use the Algo-1 [7] that is already implemented and uses the Dijkstra routing algorithm, the EON DP-QPSK modulation format, the Spectrum Sharing technique for management of the spectrum and First-Fit for the allocation of the spectrum.

When the simulations reach its end, the interface presents the results to the user, e.g. blocking ratio. Fig. 3 (a) shows the results of the 71 topologies tested, for loads from 350 to 600 Erlang. The interface allows selecting the results that will be displayed to facilitate the analyses. For instance, Fig. 3 (b) shows the results of five topologies, the original NSFNET and four topologies generated from it by the addition of a link. It is noticeable that the addition of a link impacts the network performance, e.g., the blocking rate, either positively or negatively.

IV. CONCLUSION AND FUTURE WORK

In this paper, we present a graphical interface for ElasticO++, an Elastic Optical Network Simulation Framework for OMNeT++. The implemented interface facilitates and enhances the use of ElasticO++, offering a new execution form that increases its usability, and allows to simulate a large set of networks simultaneously. Therefore, it increases the tool's potential to assist users in their research projects, providing a set of tools to implement, test, and analyze the results of
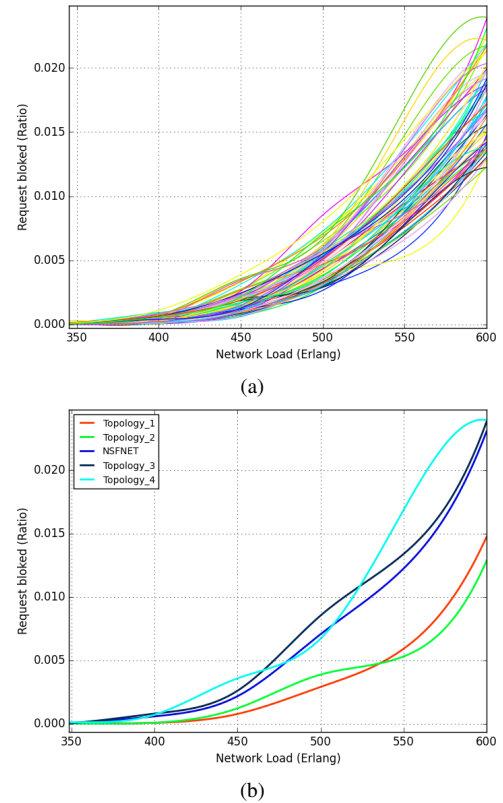

(a)

(b)

Fig. 3. Blocking results for the set of topologies. (a) All topologies. (b) Selected topologies and the original NSFNET.

new algorithms and scenarios. As future work, we intend to continue developing new features, adding new controls, and implementing new algorithms in the tool.

REFERENCES

[1] Chatterjee, Bijoy Chand; BA, Seydou; and Oki, Eiji. Fragmentation problems and management approaches in elastic optical networks: a survey. IEEE Communications Surveys & Tutorials, v. 20, n. 1, p. 183-210, 2018.

[2] Chatterjee, Bijoy Chand, Nityananda Sarma, and Eiji Oki. "Routing and spectrum allocation in elastic optical networks: A tutorial." IEEE Communications Surveys & Tutorials 17.3 1776-1800, 2015

[3] Gerstel, O., Jinno, M., Lord, A., and Yoo, S. B. Elastic optical networking: A new dawn for the optical layer?. IEEE Communications Magazine, 50(2), s12-s20, 2012

[4] Wang, Rui, and Biswanath Mukherjee. "Spectrum management in heterogeneous bandwidth optical networks." Optical Switching and Networking 11 (2014): 83-91.

[5] Chatterjee, Bijoy Chand, and Eiji Oki. "Performance evaluation of spectrum allocation policies for elastic optical networks." 17th international conference on transparent optical networks (ICTON). IEEE, 2015.

[6] Tessinari, R. S., Paiva, M. H., Monteiro, M. E., Segatto, M. E., Garcia, A. S., Kanellos, G. T., Nejabati, R., and Simeonidou, D. (2018, December). On the Impact of the Physical Topology on the Optical Network Performance. In 2018 IEEE British and Irish Conference on Optics and Photonics (BICOP) (pp. 1-4). IEEE.

[7] Tessinari, R. S., Puype, B., Colle, D., and Garcia, A. S. (2016). ElasticO++: An elastic optical network simulation framework for OMNeT++. Optical Switching and Networking, 22, 95-104.

[8] Varga, A., OMNeT++. Available at https://omnetpp.org/.