Towards simulating dynamic routing and wavelength assignment using GNPy and SIMON

Boyang Hu and Byrav Ramamurthy
Dept. of Computer Science & Engg, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, USA
Email: {bhu,byrav}@cse.unl.edu

2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) | 978-1-6654-4893-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ANTS52808.2021.9936914

***Abstract*—In this article, we enhanced the capability of SIMON (Simulator for Optical Networks) by considering the non-linear effect of the optical network components and different industry network devices. This is achieved by using an optical route planning library called GNPy (Gaussian Noise model in python) as the calculation model within SIMON. SIMON is implemented in C++ and has mainly been used as an optical network learning tool for studying the performance of wavelength-routed optical networks. It measures the network blocking probability by taking into consideration the optical device characteristics. SIMON can capture the most significant impairments when estimating the Bit-Error Rate (BER) but does not consider fiber dispersion and non-linearities. These impairments can be significant when simulating a large-scale network. GNPy, on the other hand, considers those physical impairments and can give a more accurate signal-to-noise ratio (SNR) estimation validated by real-world measurements. By integrating GNPy with SIMON, we are able to set a minimum SNR threshold, which must be satisfied by any call set up in the network. The integration of SIMON and GNPy makes the resulting simulator not only suitable for academic learning but also valuable for real-world network planning, evaluation, and deployment of optical networks.**

***Keywords*—Optical networks simulation, WDM, performance measurement, BER, Gaussian-noise (GN) model**

I. INTRODUCTION

Optical networks are the most used technology for implementing the telecommunications carriers' backbones. There are three categories for optical networks, opaque, transparent, and all-optical networks [1]. Among these three, all-optical networks are the least expensive ones that are used most in reality. The main reason is that they do not require optical-electronic-optical (O/E/O) conversions at intermediate nodes, which eliminates the cost of the regenerators. Yet, all-optical networks suffer from signal degradation during transmission because of no signal regeneration along the light path, which would end up with the low quality of transmission (e.g., leading to high bit-error rates) [2], [3]. Thus, accurately estimating the degradation of the optical signal along the possible lightpaths becomes a fundamental problem while implementing the all-optical networks.

There are mainly three strategies to address the estimation problem: numerical calculation, optical testbed measurements, and network simulation. The numerical techniques calculate the wave propagation effects in optical fibers and devices. This method is relatively accurate but impracticable for evaluating large networks due to the high computational complexity. The optical testbed option is costly and lacks flexibility due to the high cost of the optical devices deployed in the networks. Compared with the previous two methods, network simulation proves to be a good alternative in estimating network performance. It uses computers to simulate the network operation. Network designers/operators can use computational simulations with simplified analytical models. It is also highly flexible as users can quickly implement, test, and evaluate their protocols or algorithms.

There are several open-source tools for studying optical networks [4]–[7]; these tools have not been able to take into account the impact of non-linearities directly. They do not model specific vendor product parameters and thus are not amenable for real-world validation of optical networks. GNPy [8] is a newly developed optical network route planning and optimization tool that models real-world mesh optical networks. It is based on the Gaussian Noise Model [9]. One limitation of GNPy is that the simulation is stateless, and it can only handle calls one at a time. This paper addresses the challenges of using GNPy to model and simulate dynamic routing and wavelength assignment (DRWA) in optical WDM networks, where calls are admitted based on an SNR threshold.

This paper is organized as follows: we first discuss the background of SIMON and GNPy; then we explain our hybrid simulation approach using SIMON and GNPy simulator; we present the simulation results on different network scenarios in the Simulation Results section, and finally, we discuss our findings and give conclusions.

II. BACKGROUND

SIMON [4] is an object-oriented event-driven simulation package implemented in C++. It is capable of measuring the network-level blocking probability of WDM optical networks.

By incorporating WDM, the optical network can exploit the enormous bandwidth in an optical fiber. Multiple channels can be operated on different wavelengths simultaneously on a single fiber. It allows the user to adjust the parameters of the optical devices, choosing from different routing and wavelength assignment (RWA) algorithms that are to be used in the simulation.

The physical-layer models in SIMON consider signal attenuation in fiber and other components, amplifier gain saturation, and homowavelength crosstalk in switches. Simulation experiments can be performed with a user-specified bit-error-rate limit, which must be satisfied by any call set up in the

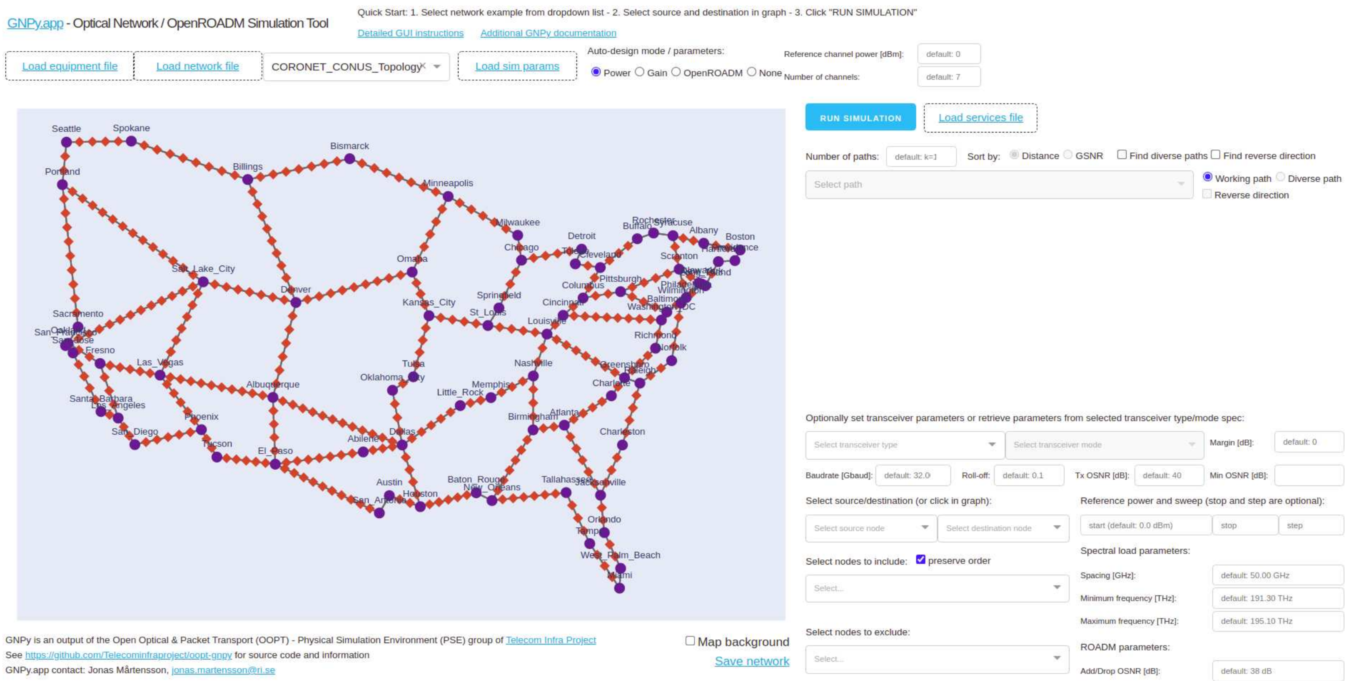

Fig. 1: GNPy web application interface

```
Result summary
req id   demand                                  snr@bandwidth A-Z (Z-A)   snr@0.1nm A-Z (Z-A)   Receiver minOSNR   mode     Gbit/s
     nb of tsp pairs      N,M or blocking reason
0       trx Lorient_KMA to trx Vannes_KBE :         24.62                  28.7                 14             mode 1     100.0
           1                   (-284,4)
1       trx Brest_KLA to trx Vannes_KBE :           17.59                  21.68                14             mode 1     200.0
           2                   (-272,8)
3       trx Lannion_CAS to trx Rennes_STA :         22.21                  26.29                13             mode 1      60.0
           1                   (-284,4)
4       trx Rennes_STA to trx Lannion_CAS :         15.94                  23.16                17             mode 2     150.0
           1                   (-258,6)
5       trx Rennes_STA to trx Lannion_CAS :         20.22                  27.44                17             mode 2      20.0
           1                   (-274,6)
7 | 6   trx Lannion_CAS to trx Lorient_KMA :        19.24                  23.32                14             mode 1     700.0
           7                  (-224,28)
7b      trx Lannion_CAS to trx Lorient_KMA :        19.32                  23.4                 14             mode 1     400.0
           4                  (-172,24)
Result summary shows mean SNR and OSNR (average over all channels)
(gnpy) ubuntu@gnpy:~/oopt-gnpy$
```

Fig. 2: GNPy terminal interface

network. However, we use the ideal physical layer for SIMON simulations in this study.

GNPy [9], [10] is an open-source route planning and optimization tool for real-world mesh optical WDM networks. It implements the Gaussian Noise Model. Its core engine is a quality-of-transmission estimator for coherent WDM optical networks.

The most appealing feature of GNPy is the accuracy of its physical impairment model [8]. It considers not only the active NEs (Network Elements) such as amplifiers and ROADMS (Reconfigurable Optical Add/Drop Multiplexers) but also the passive ones, like the optical fiber. GNPy also considers the noise introduced by the amplifiers during the signal propagation through the network. It accurately describes the EDFA (Erbium-Doped Fiber Amplifier) and its noise characteristics in terms of the Noise Figure (NF) of an amplifier model as a function of its operating point.

The GNPy software is versatile. It can be used as an engine of what-if analysis on the physical layer, optimizing the network configuration to maximize the channel capacity, and investigating the capacity and performance of a deployed network. Researchers have validated GNPy by feeding it with data from the network controller and comparing the results to experimental measurements on mixed-fiber, Raman-amplified, multi-vendor scenarios over the full C-band [8].

GNPy offers two simulation modes. One is through their web application (see Fig. 1), the other as a local application (e.g. running in a Python environment on Ubuntu) (see Fig. 2).

III. Architecture

We explain the hybrid simulation approach using the block diagram in Fig.3. First, the SIMON simulator generates calls between the source and destination pairs. These calls also contain a holding time when first initialized according to some distribution and will be released once the time expires. For each call request, the event-driven simulation module will

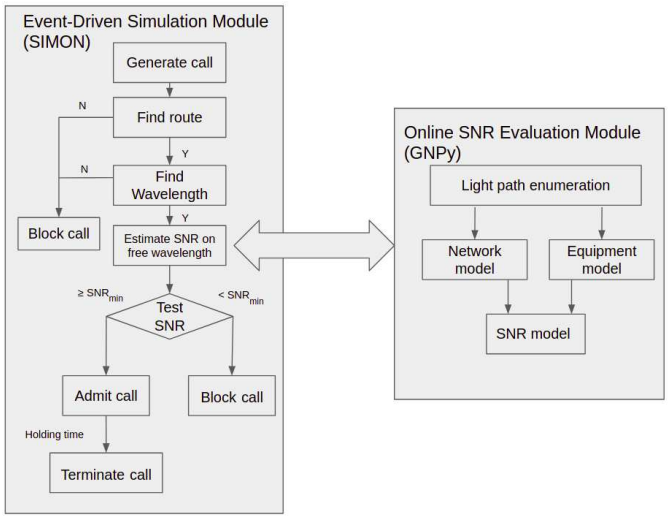

Fig. 3: Hybrid simulation technique

Fig. 4: 30 Nodes CONUS Backbone Network Topology

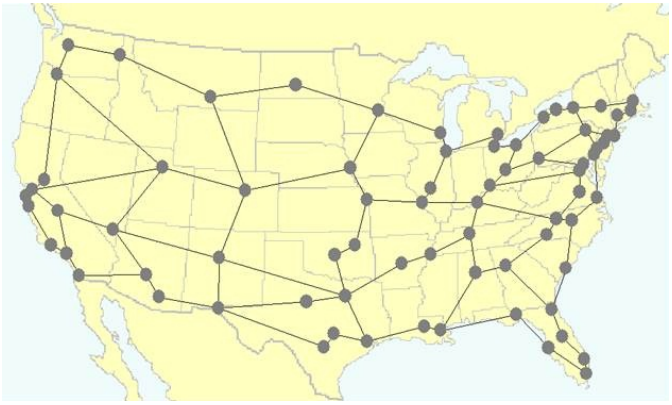

Fig. 5: 75 Nodes CONUS Backbone Network Topology

first check if there is a route available between the source and destination and then look for a free wavelength. The shortest-path routing algorithm determines the route, and a free wavelength is assigned based on the first-fit method. If there is no route or wavelength available at the time, the call will be blocked; otherwise, the light path is identified, and the simulation is switched over to the online SNR evaluation module, where the GNPy comes in. GNPy will calculate the estimated SNR for each path request based on its physical model, which considers the amplified spontaneous emission (ASE) noise, nonlinear interference (NLI) accumulation, fiber dispersion, and some other physical impairments. GNPy will send back the SNR estimate at the receiver to SIMON. If the SNR is above the pre-determined threshold (e.g., 15 dB), the call will be admitted; otherwise, the call is blocked.

The blocking probability is given by:

$$P_b = \frac{Number\ of\ blocked\ calls}{Total\ number\ of\ calls} \times 100\% \qquad (1)$$

IV. Simulation Results

In this paper, we analyze two network topologies depicted in Fig. 4 [11] and Fig. 5 [11]. These refer to a Continental United States (CONUS) 30 Nodes network topology [12] which comprises of 30 nodes and 36 links, with an average node degree of 2.4 and a CONUS 75 Nodes network topology [13] that consists of 75 nodes and 99 links, with an average node degree of 2.6. For both GNPy and SIMON, we applied the same network settings, set the OSNR threshold to 14 dB, and ran several experiments with different call requests (1,000 and 10,000 calls, respectively). For each experiment, we allocated ten wavelengths in total for the network.

SIMON has four different blocking reasons: SOURCE BUSY, DESTINATION BUSY, NO FREE WAVELENGTH, and NO ROUTE. For GNPy, we add one more constraint that is BELOW SNR THRESHOLD. The simulation results are shown in the following tables.

For the 30 nodes CONUS network, in the first experiment, the total blocked calls in SIMON is 69 and in SIMON+GNPy is $69 + 166 = 235$ as shown in Table I. The blocking probability in SIMON only and SIMON+GNPy is 6.9% and 23.5%, respectively. In our second experiment, we initiated $10,000$ calls and presents the results in Table II. SIMON blocked 795 calls due to No Free Wavelength. The GNPy blocked 240 calls because the destination received OSNR is below the threshold.

TABLE I: Reasons of blocking calls CONUS 30 (a)

1,000 calls	SIMON	SIMON + GNPy
SOURCE_BUSY	0	0
DESTINATION_BUSY	0	0
NO_FREE_WAVELENGTH	69	69
NO_ROUTE	0	0
BELOW_SNR_THRESHOLD	N/A	166

For the 75 nodes CONUS network, in our first experiment, the total blocked calls in SIMON is 53 and in SIMON+GNPy is $53 + 240 = 293$ as shown in Table III. The blocking probability in SIMON and SIMON+GNPy is 5.3% and 29.3%, respectively. Our third experiment results are presented in Table

TABLE II: Reasons of blocking calls CONUS 30 (b)

10,000 calls	SIMON	SIMON + GNPy
SOURCE_BUSY	0	0
DESTINATION_BUSY	0	0
NO_FREE_WAVELENGTH	795	795
NO_ROUTE	0	0
BELOW_SNR_THRESHOLD	N/A	240

IV. The blocking probability in SIMON and SIMON+GNPy is 5.75% and 17.62%, respectively. The low OSNR quality causes additional blocked calls from GNPy at the destination node.

TABLE III: Reasons of blocking calls CONUS 75 (a)

1,000 calls	SIMON	SIMON + GNPy
SOURCE_BUSY	0	0
DESTINATION_BUSY	0	0
NO_FREE_WAVELENGTH	53	53
NO_ROUTE	0	0
BELOW_SNR_THRESHOLD	N/A	240

TABLE IV: Reasons of blocking calls CONUS 75 (b)

10,000 calls	SIMON	SIMON + GNPy
SOURCE_BUSY	0	0
DESTINATION_BUSY	0	0
NO_FREE_WAVELENGTH	575	575
NO_ROUTE	0	0
BELOW_SNR_THRESHOLD	N/A	1187

We also use GNPy to compute the destination SNR in each unique path based on different shortest path algorithms (shortest hop path and shortest length path, which obtained from SIMON) and compare the behavior between these two algorithms in terms of blocking probability. The results are shown in Table V.

For CONUS 30 and CONUS 75 network topology, there are 870 and $5,550$ unique paths, respectively. The shortest length path algorithm performs better than the shortest hop path algorithm in terms of blocking probability at the destination receiver. This is mainly because, in the shortest length path algorithm, the light path traverses less physical distance than in the shortest hop path algorithm, thus less power loss at the destination.

TABLE V: Reasons of blocking calls in CONUS 30 & 75

CONUS 30	Shortest Hop	Shortest Length
BELOW_SNR_THRESHOLD	240	217
CONUS 75	Shortest Hop	Shortest Length
BELOW_SNR_THRESHOLD	1418	1203

V. Discussion

Our simulation results show that by incorporating GNPy into SIMON, the blocking probability has increased. This is expected because GNPy considers the physical impairments and non-linearity effects of network components which can accurately model the real network situation.

In this work, we addressed the challenge of integrating SIMON and GNPy and automating the simulation process. We suggest that the GNPy developer community consider implementing the stateful simulation feature as this would be greatly useful when simulating dynamic routing and wavelength assignment with multiple calls. The effects of an active call on another potential call can be modeled more accurately with a stateful implementation of GNPy.

VI. Conclusions

We present a new hybrid simulation approach for the optical network using both SIMON and GNPy. By leveraging the physical impairment modeling in GNPy, we make SIMON not only suitable for learning purposes but also able to simulate the real network environment and achieve a more accurate blocking probability estimation. We plan to further integrate SIMON and GNPy and explore machine learning applications using the GNPy simulator in our future work.

VII. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number CNS-1817105.

References

[1] B. Ramamurthy, H. Feng, D. Datta, J. P. Heritage, and B. Mukherjee, "Transparent vs. opaque vs. translucent wavelength-routed optical networks," in *OFC/IOOC. Technical Digest. Optical Fiber Communication Conference, 1999, and the International Conference on Integrated Optics and Optical Fiber Communication*, vol. 1. IEEE, 1999, pp. 59–61.

[2] B. Ramamurthy, D. Datta, and H. Feng, "Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks," *Journal of Lightwave Technology*, vol. 17, no. 10, p. 1713, 1999.

[3] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J. S. Pareta, and I. Tomkos, "A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks," *Computer networks*, vol. 53, no. 7, pp. 926–944, 2009.

[4] B. Ramamurthy, D. Datta, H. X. Feng, J. P. Heritage, and B. Mukherjee, "SIMON: A simulator for optical networks," in *All-Optical Networking 1999: Architecture, Control, and Management Issues*, vol. 3843. International Society for Optics and Photonics, 1999, pp. 130–135.

[5] P. Pavon-Marino and J.-L. Izquierdo-Zaragoza, "Net2plan: an open source network planning tool for bridging the gap between academia and industry," *IEEE Network*, vol. 29, no. 5, pp. 90–96, 2015.

[6] M. A. Cavalcante, H. A. Pereira, and R. C. Almeida, "SimEON: an open-source elastic optical network simulator for academic and industrial purposes," *Photonic Network Communications*, vol. 34, no. 2, pp. 193–201, 2017.

[7] D. A. Chaves, H. A. Pereira, C. J. Bastos-Filho, and J. F. Martins-Filho, "Simton: A simulator for transparent optical networks," *Journal of Communication and Information Systems*, vol. 25, no. 1, 2010.

[8] A. Ferrari, M. Filer, K. Balasubramanian, Y. Yin, E. Le Rouzic, J. Kundrát, G. Grammel, G. Galimberti, and V. Curri, "GNPy: an open source application for physical layer aware open optical networks," *Journal of Optical Communications and Networking*, vol. 12, no. 6, pp. C31–C40, 2020.

[9] Telecom Infra Project - OOPT PSE Group. GNPy documentation. (April 07, 2021). [Online]. Available: https://gnpy.readthedocs.io/_/downloads/en/master/pdf/

[10] GNPy source code. (April 07, 2021). [Online]. Available: https://github.com/Telecominfraproject/oopt-gnpy

[11] "Sample optical network topology files," http://www.monarchna.com/topology.html, accessed: 2021-07-01.

[12] J. M. Simmons, *Optical network design and planning*. Springer, 2014.

[13] D. W. Matula and R. R. Sokal, "Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane," *Geographical analysis*, vol. 12, no. 3, pp. 205–222, 1980.