Modelling Optical Network Components: A Network Simulator-Based Approach

Vedran Miletić
University of Rijeka Department of Informatics
Omladinska 14, 51000 Rijeka, Croatia
vmiletic@inf.uniri.hr

Branko Mikac
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
branko.mikac@fer.hr

Matija Džanko
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
matija.dzanko@fer.hr

***Abstract*—The paper describes the design goals and methodology in creating a new model of optical telecommunication network. The model is implemented by discrete-event network simulator ns-3. The advantages of using the existing simulator core infrastructure provided by ns-3 are analyzed and compared to building own simulator from scratch, or selecting a tool among other existing simulators such as ns-2, OMNeT++, and OPNET. The requirements for feature functionality are outlined and high-level overview of the model architecture and its components are provided. Finally, the possibilities for extending the model in future research and development work are described.**

***Index Terms*—optical networks, modelling, simulation, reliability**

I. Introduction

Optical wavelength division multiplexing (WDM) technology allows partitioning the large available bandwidth into a number of smaller channels. The technology is advancing rapidly, the number of channels is increasing and this has made and it one of the key parts of future optical network environments.

To achieve optimal working of optical WDM networks, considerable research activity is needed. Simulation can help here by providing researchers with a cost effective method to study and compare the behavior of proposed algorithms. However, a lack of single uniform simulation platform for optical WDM networks makes it very difficult for researcher and engineers to compare results, since model specifics of different simulators can lead to significant differences in results. Furthermore, disparate sets of features provided by different simulators and lack of integration usually limit research possibilities.

To address this issue, a simulator named OWns [1], [2] was developed by extending ns-2 simulator. OWns models key features of WDM networks, including optical switching nodes, multi-wavelength links, routing and wavelength assignment (RWA) algorithms.

A simulation tool is required for research purposes that can extended into various fields of optical network research as needed. We found that none of the existing simulators, aside from OWns and some commercial solutions, had any of the required feature functionality. We decided to avoid commercial solutions due to reasons we describe in detail below, and we found OWns to be based on outdated simulation platform with limited extensibility.

We developed a new model for optical network in network simulator ns-3 and named it PWNS (short for prototype WDM network simulator, name chosen when we started developing it[1]). We explain below why have we selected ns-3 among other available simulation platforms.

The rest of the paper is organized as follows. Section 2 compares available simulation software. Section 3 lists requirements for optical WDM network simulator. Section 4 gives high-level overview of our WDM network simulator design. Section 5 describes the simulation of a circuit-switched optical network in case of failures. Section 6 concludes the paper and lists possible directions for future work.

II. Network simulation software

We didn't consider proprietary network simulators due to limited use conditions, limited extensibility, and also license cost. We wanted a solution that is extensible and freely available to researchers and engineers working in various subfields of optical network research so that they can extend the model to their needs.

We evaluated the possibility of developing own simulator, in terms of feature functionality somewhat similar to COSMOS [3] but more extensible. In that case we would simulate network layers from L3 to L1. But, as we would like to leave open the possibility of researching complex networks that aren't just optical on the physical layer, we decided to extend an existing open source network simulator.

We thoroughly analyzed open source network simulators ns-2 [4], OMNeT++ [5] and ns-3 [6] [7].

Funding of ns-2 development has decreased a lot in the last decade, and this has resulted in decreased integration of additional models developed by network researchers into mainline ns-2. Today there are many incompatible (and therefore incomparable) models with various features that can be

[1]Since "pwn" is a leetspeak slang term derived from the verb "own", the name PWNS is a pun on OWns in addition to this.

found on the Internet, and many of them also depend on specific version of ns-2 so it's not realistic to expect integration into mainline anytime soon.

Aside from this, ns-2 has a bunch of other design limitations [8]:

- split object model (OTcl and C++),
- relatively high amount of abstraction in network layer and below increases the difficulty in connecting simulation and real world,
- lack of additional simulation tools, for example, steady-state simulation detector,
- lack of model validation, and
- lack of model documentation.

Despite the fact that we didn't consider using and extending ns-2 because of the limitation listed and expectation that network research community will mostly transition to ns-3 in next couple of years, it's worth mentioning OWns [1] [2] variant that implements WDM network model. OWns is no longer developed and its source code is no longer officially available.

Our criterium, aside from having a model for optical transmission network and support for network reliability analysis, was simulator architecture and its performance. There is passive optical network model for OMNeT++ [9]. But this model is not applicable because it's model of access network, and our research interest is in core network. Aside from that, OMNeT++ has some of the drawbacks of ns-2: component model is similar [10], its architecture is bilingual (using NED and C++), and it's tightly integrated in its IDE, implemented as an Eclipse IDE plugin. In addition to that, OMNeT++ uses Academic Public License, that prohibits commercial use. We consider this a major obstacle because it eliminates companies that could potentially be interested in developing and further extending the model we propose in the following text.

Despite the fact that among these three simulators ns-3 has demonstrated best overall performance [11], both ns-3 and OMNeT++ are capable of carrying out large-scale network simulations in an efficient way, with ns-2 exhibiting longer simulation runtime in simulations consisting of a few thousand simulation nodes.

ns-3 network simulator was designed and written from scratch. As shortcomings of ns-2 come largely from its design decisions, it was impossible to resolve them and at the same time keep compatibility with existing simulator core and already developed models. During ns-3 development ideas and parts of code were taken from GTNetS, yans [12], and ns-2 simulators. The development was supported by National Institute for Research in Computer Science and Control (Institut national de recherche en informatique et en automatique, INRIA) and National Science Foundation. The goal of the project was to create a tool that will be developed by the academic community and companies even after the initial funding dries up. To achieve that, ns-3 Project created a community of maintainers, people responsible for a certain part of simulator code. In addition, the infrastructure was set up so that any interested person can join the development, either by further developing existing models, or by creating new ones [13]. The entire ns-3 code is available under GNU General Public License, version 2.

ns-3 simulator is based on discrete events. Simulated time is represented using integer type to avoid problems with portability on different processor architectures and operating systems [10]. Size of data type that is used to represent a moment in simulated time is 128 bit; 64 bit is used for integer, and 64 bit for floating-point part. This allows simulating 584 years with nanosecond precision.

ns-3 contains pseudorandom number generator MRG32k3a [14]. MRG32k3a generator offers $1.8 \cdot 10^{19}$ independent sequences of random numbers, each containing $2.3 \cdot 10^{15}$ subsequences. Each subsequence has period $7.6 \cdot 10^{22}$. Period of the entire generator is $3.1 \cdot 10^{57}$. Other random number generators, such as Mersenne twister [15] with total period $2^{19937} - 1$, can be used if necessary.

Simulator is single-threaded by design. Multithreaded simulator was evaluated, but because smart pointers are used for automatic garbage collection, it happens that multithreaded variant performs slower than singlethreaded [16]. Our primary interest is reliability analysis of optical telecommunication network, where one has the option to run multiple independent simulations, so this isn't a problem.

Up until now, ns-3 lacked a model for optical transport network components model. Due to the fact that none of the models contained in other simulators was found to be adequate and flexible architecture of ns-3 simulator it was decided that we will develop our own optical network model and build tools for reliability analysis based on that. Concepts and ideas in already mentioned solutions for other two simulators can be useful as a pointer in certain direction.

III. Model requirements

Given the present feature functionality of ns-3 network simulator, we had to identify the specific areas where it was to be extended to support simulating optical WDM network. We considered the differences between the networks that have existing models in ns-3, which operate almost entirely in the electronic domain, and the optical WDM networks, which operate in both the optical and electronic domain.

Some of the requirements outlined in [1] apply to almost any optical network simulator. Specifically, the following is required:

- **Multi-wavelength Channels:** Optical WDM technology uses multiple wavelengths for data transmission over a fiber link. The support is needed for both coarse and dense WDM, and also both unidirectional transmission channels and bidirectional transmission channels.
- **Optical Switch Devices:** Models for devices in the optical network should include devices that act as switches with varying degrees of wavelength conversion capabilities.
- **Switching Granularity:** The model has to support various degrees of switching granularity, for example

switching at fiber level, at wavelength level and at sub-wavelength level [17].

- **Switching Paradigms:** Model has to support Optical Circuit Switching (OCS) and allow the implementation of other switching paradigms such as Optical Burst Switching (OBS).
- **Switching Architectures:** The model has to allow detailed specification of interconnections of switch device parts.
- **Control Plane:** The model has to support a control plane (for routing, resource reservation, failure recovery etc.), ideally by reusing already existing solution such as OpenFlow [18].

Once these components are implemented, network research and engineering in general in the field of optical WDM networks can be studied by using simulation inside a network simulator. This is specifically interesting to subfields such as multilayer recovery[19], where the possibility of using simulation heavily depends on having optical WDM network model implemented inside a network simulator such as ns-3, since it implements entire layer stack.

IV. Modelling the optical transmission network

Inherent similarity between models of various types of telecommunication networks (including optical networks) suggests that approach that involves adding reusable features or functional parts to a component is more appropriate than implementation of the whole solution in one large monolithic model at once. Model based on components is easier to develop, test, verify and validate, because the implementation of feature functionality can happen iteratively, first implementing a feature and then testing the implementation. In addition, code reusability inherent in object-oriented design reduces the time needed to develop a similar model.

Components of ns-3 are modules, which consist of one or more classes which together make one or more models of real world communication devices, communication channels, network protocols etc. Abstract base classes used implemented by every model of a physical network in ns-3 are `NetDevice` and `Channel`. `NetDevice` describes a network interface card at a network node; `Channel` interconnects two or more network cards and contains delays, losses etc. Models of complex networks (e.g. WiFi, WiMAX, LTE) usually separate PHY layer of the network card from its MAC layer to allow combining various MAC devices and PHY devices and facilitate code reuse.

All the classes that implement the model of optical telecommunication network are contained in `optics` module and have names that begin with `Wdm` for consistency. We describe the most used ones.

`WdmNetDevice` and `WdmChannel` are abstract base classes that have features common to all optical network devices and channels. For network devices, this includes receive error model, lists of physical interfaces and elements needed by ns-3. For channel, this includes propagation loss and delay models.

`WdmNetDevice` class is used by `WdmPassthroughNetDevice` abstract base class and `WdmEdgeNetDevice`. `WdmEdgeNetDevice` is network device used at the edge of optical network that transmits and receives optical signal and does the conversion from/to bits and bytes. `WdmPassthroughNetDevice` is used as a base class for classes that model behavior of optical network devices that signal passes through in some way. Examples of such devices are multiplexers (`WdmMuxNetDevice` class), demultiplexers (`WdmDemuxNetDevice` class), add-drop multiplexers (`WdmAdmNetDevice` class) and optical cross-connects (`WdmOxcNetDevice` class). Class hierarchy can be seen in figure 1.

All of these devices share common code for physical interfaces, implenented in `WdmPhy` class and it's subclasses `WdmInputPhy` and `WdmOutputPhy`, modelling physical reception and transmission interfaces respectively. A diagram representation of a simple example that shows the relation between physical interfaces, network devices and channels can be seen in figure 2.

`WdmPhy` implements both ITU-T DWDM [20] and CWDM [21] grids. DWDM grid is used by default. 64-bit integer type is used internally to represent signal frequency (in Hz) instead of floating point type for wavelength (in nm). The argument for doing this is the same as the argument for using integer type to represent simulation time in ns-3; we wanted to avoid problems with floating-point type with regards to portability. Application programming interfaces for working both with frequencies and wavelengths inside of a simulation are provided, so users and model developers can use one they see fit for the problem at hand.

`WdmChannel` class is used by `WdmUnidirectionalChannel` class that models fiber used for transmission in a single physical direction. It is expected to have `WdmInputPhy` at one end and `WdmOutputPhy` at other end. Physical effects other than loss and delay (chromatic dispersion, scattering, four wave mixing etc.) are not modelled yet, but the placeholders are left in the model for future implementation of these features.

Since our primary interest is in the field of reliability, we evaluated the the existing possibilities. ResumeNet[2] project, finished in August 2011, extended ns-3 simulator to analyze network availability [22] [23]. Implementation of reliability model presented in these papers works by stopping network interface at a node in case of node failure, and network interface at both ends in case of a link failure. This is a rather simple but quite unrealistic solution. Big issue here is that model specifically uses IPv4 and can't be used in case of

[2]Resilience and Survivability for Future Networking (ResumeNet) is a collaboration between The University of Kansas (KU), Lancaster University, ETH Zürich, Techniche Universität München (TUM), Techniche Universiteit Delft, Université de Liège (ULg), Universität Passau, Uppsala Universitet (UU), NEC Laboratories Heidelberg and France Telecom – Orange Labs. ResumeNet researches framework, mechanisms and experimental evaluation of network resilience and survivability in presence of failures for future networks and is funded by EU Future Internet Research & Experimentation (FIRE) from Seventh Framework Programme (FP7).

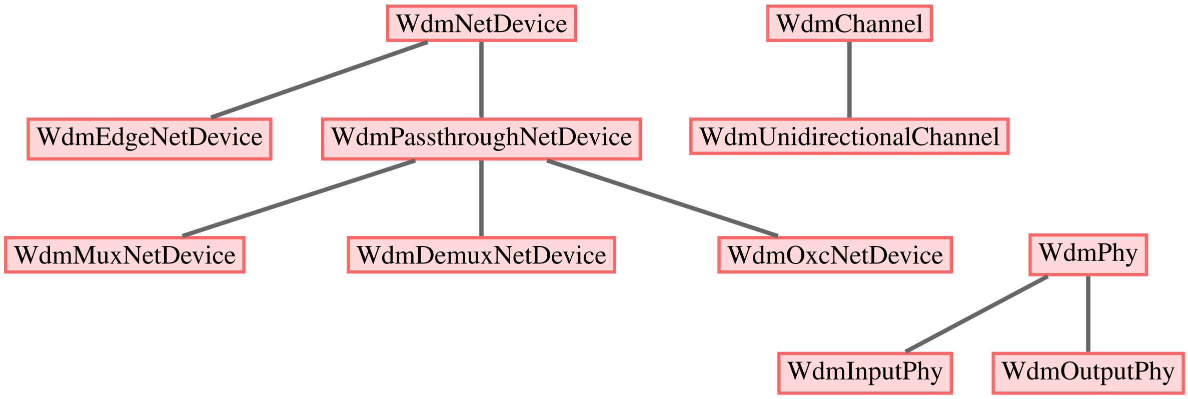

Figure 1. Class hierarchy.

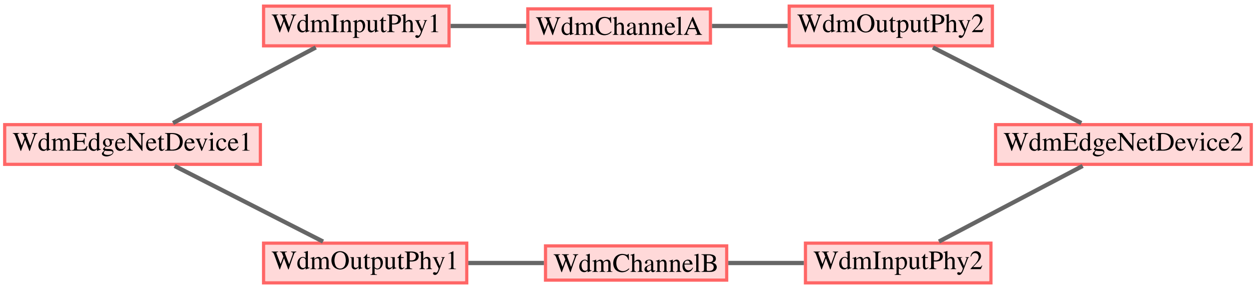

Figure 2. Relation between device, physical interface and channel.

IPv6 nodes or don't use IP at all for sending and receiving. Other frameworks developed in this project (e.g. topology and failure specification) might be usable in the future with some adaptations.

We opted to implement failure state directly in ns-3's base object class `Object`. Since mentioned classes `WdmNetDevice`, `WdmChannel` and `WdmPhy` derive from this class, this allows them to get information about current state of the object (failed or working). By default, this doesn't alter anything on existing classes in ns-3 simulator and researcher creating or extending a model has to explicitly use this feature.

Building upon this, `FailureRepairModel` is attached to `Object` and can change its state in accordance to elapsed time. It does so by calculating next event (be it failure or repair) and scheduling it to happen a certain time interval after the current simulation time. Times to failure and repair of objects are calculated during simulation runtime according to user configurable probability distributions.

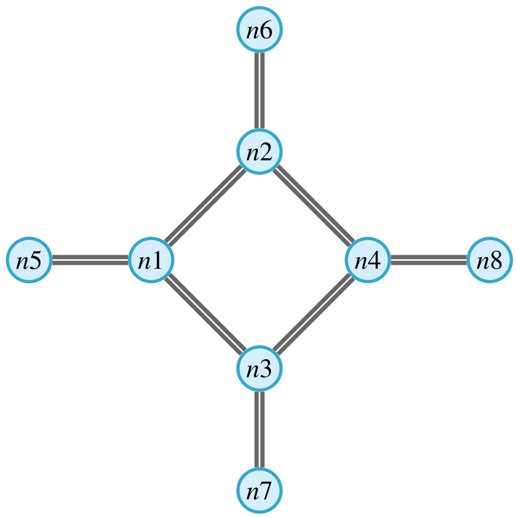

Figure 3. Topology for the case study.

V. Case Study

The simulation presented here is based on the following scenario: the network consists of four WDM OXCs which are modelled as `WdmOxcNetDevices` at nodes interconnected by pairs of fibers modelled as pairs of `WdmUnidirectionalChannels`. Each OXC is assumed to be transparent meaning that it doesn't read packet headers; it demultiplexes them based on the input wavelength, and switches them to the appropriate output link and wavelength, based on preconfigured information. OXCs are assumed not to possess any kind of wavelength conversion capabilities. Traffic generators (ns-3 on-off applications) and packet sinks are attached to the edge nodes.

The simulation can be configured by varying device and physical interface attributes such as usage of coarse vs dense WDM and number of inputs and outputs.

Consider an eight node network with physical topology as shown in the Figure 3. Dense WDM with 100Ghz channel

Mean time to failure for links	Number of packets sent	Number of packets received	Number of packets lost	Percentage of packets lost
30 days	5122922	5010003	112919	2.2%
60 days	5122922	5069166	53756	1.05%
90 days	5122922	5078451	44471	0.87%
120 days	5122922	5087854	35068	0.68%
180 days	5122922	5094886	28036	0.55%

Table I
SIMULATION RESULTS.

spacing is used and each channel has bandwidth of 10 Gbit/s. All channels have delay set to 10 ms. In this scenario, main and backup lightpaths are statically defined (in addition to this approach, lightpaths can also be computed using any of the frequently used heuristic techniques):

- In case of no failure or in case of failure of link $n3-n4$, for communication between node $n5$ and node $n8$ path over nodes $n1$, $n2$ and $n4$ is used, and for communication between node $n6$ and node $n7$ path over nodes $n2$, $n1$ and $n3$ is used,
- In case of failure of link $n1-n2$, for communication between node $n5$ and node $n8$ backup path over nodes $n1$, $n3$ and $n4$ is used, and for communication between nodes $n6$ and $n7$ path over nodes $n2$, $n4$ and $n3$ is used.
- In case of failure of link $n1-n3$, for communication between nodes $n5$ and $n8$ main path is used, for communication between $n6$ and $n7$ backup path is used.
- In case of failure of link $n2-n4$, for communication between nodes $n5$ and $n8$ backup path is used, for communication between $n6$ and $n7$ main path is used.

We run the simulation for mean time to failure values 30 days, 60 days, 90 days, 120 days, and 180 days (all exponentially distributed) for $n1-n2$, $n2-n4$, $n1-n3$ and $n3-n4$. Mean time to repair is set to 8 hours (constant). Links $n1-n5$, $n2-n6$, $n3-n7$, and $n4-n8$, as well as all nodes are assumed to be completely reliable, i.e. assumed to be unable to fail.

On-off application at nodes $n5$ and $n6$ and packet sinks at nodes $n7$ and $n8$ are used for simulating traffic flows. On-off application at $n5$ is sending packets to $n8$ at using signal frequency 190,100 Ghz (1577.03 nm wavelength), and on-off application at $n6$ is sending packets to $n7$ using signal frequency 190,200 Ghz (1576.20 nm wavelength), so data transmissions occur in parallel. Both applications send 1 Kbit/s of data in packets of 1400 bytes during "on" time which lasts 20 seconds (exponentially distributed), and then have "off" time which lasts between 1 and 3 seconds (uniformly distributed). On and off times alternate. We observed that while larger data rates (100 Mbit/s, 1 Gbit/s, 10 Gbit/s) increase simulation duration in terms of wall clock time approximately by a factor of 10^5, 10^6, and 10^7 (respectively), they don't significantly alter the percentage of packets lost due to channel failures so we opted for smaller data rate.

We measure packet loss due to link failures. Even though there are backup routes, packet loss still occurs due to one of the following factors:

- Packets in transit when the failure occurs get dropped on optical cross connect adjacent to the failed link, and
- Failures of two or more links in the same period of time (usually called dual and multiple failures respectively) which cause that no working backup path exists.

Exactly 1 year of time is simulated. Simulation results are shown in table I. Please note that the number of packets sent remains the same since on-off application configuration doesn't change between simulation runs.

VI. CONCLUSION

In this paper we presented prototype WDM network simulator based on ns-3 network simulation framework. We described why ns-3 was selected as a foundation among other network simulators, we outlined requirements for optical WDM network model, and we described the model architecture. Finally, we demonstrated the functionality doing simulation of a simple eight node network.

In comparison to OWns for ns-2, our model offers possibility of simulating failure and repair of optical links and components. Other tool we mentioned, EPON for OMNET++, has different goals compared to PWNS: it models access network, while PWNS models core network.

In ns-3's Google Summer of Code 2012 one of the projects proposed to be mentored by us was developing WDM components for ns-3. Due to a large amount of high-quality applications in other areas (Internet protocol stack, wireless networks, simulation infrastructure etc.), our proposal unfortunately wasn't selected for funding so we decided to do the work ourselves regardless.

In the future we plan to validate existing models by comparing them to real world measurements and already published simulation results. In addition to this, we plan to model a wider set of devices (we are specifically interested in reconfigurable add-drop multiplexers and various architectures for multigranular optical cross-connects), physical properties of optical waves (chromatic dispersion, scattering, four wave mixing etc.), and also other types of optical networks in addition to core (such as passive optical networks in access domain), so we will be able to compare our tool with a wider set of existing tools.

Parts of the code that get thorough validation and testing will be submitted for inclusion in mainline ns-3.

REFERENCES

[1] N. Bhide and K. Sivalingam, “Design of OWns: optical wavelength division multiplexing (WDM) network simulator,” in *proceedings of First SPIE Optical Networking Workshop*. Citeseer, 2000.

[2] B. Wen, N. Bhide, R. Shenai, and K. Sivalingam, “Optical wavelength division multiplexing (WDM) network simulator (OWns): architecture and performance studies,” *SPIE Optical Networks Magazine*, pp. 16–26, 2001.

[3] M. Lacković, R. Inkret, and B. Mikac, “An approach to an IP over WDM transmission network modelling,” in *Proceedings of the 2002 4th International Conference on Transparent Optical Networks, 2002.*, vol. 1. IEEE, 2002, pp. 82–85.

[4] S. McCanne, S. Floyd, and K. Fall, “The LBNL network simulator,” *Software on-line: http://www.isi.edu/nsnam*, 1997.

[5] A. Varga *et al.*, “The OMNeT++ discrete event simulation system,” in *Proceedings of the European Simulation Multiconference (ESM’2001)*, vol. 9, 2001.

[6] T. Henderson, M. Lacage, G. Riley, C. Dowell, and J. Kopena, “Network simulations with the ns-3 simulator,” *SIGCOMM demonstration*, 2008.

[7] ns 3 Project, *ns-3 Manual, version 3.14*. [Online]. Available: http://www.nsnam.org/docs/release/3.14/manual/singlehtml/index.html

[8] M. Lacage, “Network experimentation and simulation with ns-3,” in *Trilogy Future Internet Summer School*. Trilogy consortium, 2009.

[9] K. Kim, “Integration of OMNeT++ hybrid TDM/WDM-PON models into INET framework,” in *OMNeT++ Workshop*, 2011.

[10] M. Lacage, “Experimentation tools for networking research,” Ph.D. dissertation, Ecole doctorale Stic, Université de Nice Sophia Antipolis, 2010.

[11] E. Weingartner, H. Vom Lehn, and K. Wehrle, “A performance comparison of recent network simulators,” in *ICC’09. IEEE International Conference on Communications, 2009.* IEEE, 2009, pp. 1–5.

[12] M. Lacage and T. Henderson, “Yet another network simulator,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. ACM, 2006, p. 12.

[13] ns 3 Project, *ns-3 Model Library, version 3.14*. [Online]. Available: http://www.nsnam.org/docs/release/3.14/models/singlehtml/index.html

[14] P. L’Ecuyer, R. Simard, E. Chen, and W. Kelton, “An object-oriented random-number package with many long streams and substreams,” *Operations Research*, pp. 1073–1075, 2002.

[15] M. Matsumoto and T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[16] G. Seguin, “Multi-core parallelism for ns-3 simulator,” *INRIA Sophia-Antipolis, Tech. Rep*, 2009.

[17] L. Noirie, M. Vigoureux, and E. Dotaro, “Impact of intermediate traffic grouping on the dimensioning of multi-granularity optical networks,” in *Optical Fiber Communication Conference*. Optical Society of America, 2001.

[18] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[19] B. Puype, J. Vasseur, A. Groebbens, S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, and P. Demeester, “Benefits of gmpls for multilayer recovery,” *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 51–59, 2005.

[20] ITU-T Recommendation G.694.1, “Spectral grids for WDM applications: DWDM frequency grid,” 6 2002.

[21] ITU-T Recommendation G.694.2, “Spectral grids for WDM applications: CWDM wavelength grid,” 12 2003.

[22] E. Çetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, and J. Sterbenz, “A comprehensive framework to simulate network attacks and challenges,” in *(ICUMT), 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*. IEEE, 2010, pp. 538–544.

[23] ——, “Modelling communication network challenges for future internet resilience, survivability, and disruption tolerance: A simulation-based approach,” *Telecommunication Systems*, pp. 1–16, 2011.