SimulNet: a Wavelength-Routed Optical Network Simulation Framework

Francesco Palmieri, Ugo Fiore and Sergio Ricciardi*
Federico II University of Napoli
Via Cinthia, 5, Complesso Universitario di Monte S. Angelo, 80126 Napoli, Italy
{francesco.palmieri, ugo.fiore, sergio.ricciardi}@unina.it

Abstract

Simulation seems to be the best available alternative to the deployment of expensive and complex testbed infrastructures for the activities of testing, validating and evaluating optical network control protocols and algorithms. In this paper we present SimulNet, a specialized optical network simulation environment providing the foundation for the study and analysis of the key control plane characteristics of wavelength-routed networks. Such an environment would provide researchers with an open framework for easily exploring the evolving characteristics of WDM-routed technologies which includes developing new protocol suites or performing rapid evaluation and easier comparison of results across research efforts.

Keywords: simulation, WDM-switching, RWA, OXC

1. Introduction

Simulation plays an important role in network protocol design, providing researchers with a cost effective method to analyze and study the behavior of proposed protocol models. Particularly, simulation becomes an indispensable tool in WDM-based transparent optical networking research, since it helps researchers to quickly and inexpensively validate and evaluate the performance of new protocols and algorithms without the need of installing and managing complex network testbeds that require very expensive optical devices and communication infrastructures. In fact, even on a fully featured testbed, not all algorithms, mechanisms and protocols can be readily implemented, tested and evaluated, because most of the available optical network equipments are proprietary vendor products featuring a very low degree of programmability. Implementing new control plane features and algorithms on top of such proprietary equipment can be a very tedious and difficult task, since it requires accessing, controlling and programming the low level capabilities of these devices. Unfortunately only a very limited number of simulation tools are available for conducting research related to different routing and wavelength assignment algorithms (RWA), topology management (e.g. converter placement algorithms), re-optimization, centralized and distributed wavelength reservation schemes, and the effect on RWA algorithms of inaccuracies in network-wide state information or physical-level impairments. Furthermore, most of the available solutions are limited in scope, difficult to use or not totally open. In many cases, they have also been based on simulation models designed specifically to cope only with a specific problem. Since each of these solutions used their own simulation platform, model and assumptions, it is generally difficult to reuse existing protocol modules and compare simulation results under a common assessment environment. Our aim is to address these problems with the introduction of a new simulation framework developed specifically for the testing and performance analysis of distributed dynamic RWA algorithms on optical networks and also for optimization algorithms and protocols validation. The framework we present, called *SimulNet*, has been designed to be easy to use and exhibits satisfactory performance also when simulating large network topologies. It has been implemented according to a modularized, platform-independent, and extensible architecture and provides a useful baseline library of commonly used RWA algorithms and signaling schemes. We based our framework on a totally flexible network model, supporting heterogeneous WDM equipment, with or without wavelength conversion capability, in which the number and type of lambdas can vary on each link. It provides a fully dynamic and configurable path selection scheme supporting sub-wavelength bandwidth allocation (grooming). Furthermore, it allows simulations to be aware of all the complexity, expensiveness, performance and resource-limitation constrains implicit in the various flavors of optical switching devices, for example explicitly and proportionally penalizing, when instructed to do so, all the paths that require wavelength conversion. The simulator has been successfully validated by implementing some well-known algorithms, and comparing the results with those available in literature.

* *Sergio Ricciardi is PhD student at the Department of Computer Architecture of the Technical University of Catalonia (UPC), Jordi Girona 3, 08034 Barcelona, Catalunya, Spain.*

978-1-4244-4671-1/09/$25.00 ©2009 IEEE

2. Related work

At the state of the art, several simulation packages and tools modeling optical network infrastructures are available, but none of them totally provides the necessary support and flexibility needed for the study and the performance assessment of new routing, signaling and wavelength management schemes on modern WDM-empowered networks. GLASS (GMPLS Lightwave Agile Switching Simulator) [1], developed at NIST, is an optical network simulation tool built on the SSFNET Framework [2]. It supports many advanced features, including GMPLS and QoS, and several optical failure recovery schemes. Since GLASS is implemented in Java it is easy to configure, modular and platform independent. Unfortunately, its complexity adversely conditioned its performance and hindered the development of new routing schemes on this platform. OWNS [3] is tool that is being used for research in the optical network domain. It is built on the well known mature foundation of the NS-2 [4] simulation infrastructure with a set of additional optical WDM extensions. Unfortunately, with these benefits it also inherited some inconveniences of NS-2, such as bloated code base, slow execution speed and large memory footprint. Also, configuring and implementing any new routing scheme or protocol requires extensive and careful coding, because of the absence of a modular structure. OWNS also lacks the feature set required for the implementation of converter placement algorithms. JAVOBS [5] is a set of flexible and configurable java libraries dedicated to the simulation of Optical Burst Switched (OBS) networks. It supports simulation of various OBS schemes, reservation protocols and scheduling algorithms, on any topology of reasonable size. SimulNet can be considered quite similar to JAVOBS as for the general simulation framework and the configurability, with the fundamental difference that while JAVOBS operates over OBS networks, SimulNet is expressly designed to operate over wavelength switched networks' control plane layer.

3. Basic Architecture

Simulation environments allow the user to predict the behavior of a set of network devices on a complex network, by using an internal model that is specific to the simulator. Simulators do not necessarily reproduce the same sequence of events that would take place in the real system, but rather apply an internal set of transformation routines that brings the simulated network to a final state that is as close as possible to the one the real system would evolve to. To accomplish this task, simulators work on a model that contains all the relevant information that must be observed, while abstracting irrelevant details, thus simplifying both the simulation and the analysis of the network. This approach typically allows the simulated network to scale well in size and complexity, as irrelevant details may be totally abstracted or represented in a simplified manner. Thus, the simulator can be used to manage also complex networks with hundreds or thousands of network elements and wavelengths, while greatly reducing the probability of introducing programming bugs. The drawback is that the simulated devices may have limited functionalities and their behavior may not closely resemble that of real world devices. This is the reason why both the parameters to be represented or abstracted in the model and the representation detail must be carefully chosen. An excessive degree of sophistication in the simulation environment is often the cause of a limited flexibility in configuration and introduction of new functional modules, and in general heavily taxes both the runtime performance and the ease of use. In fact, the real objective of a simulation is to make available a sustainable and effective model of the involved system that is sufficiently accurate to reliably analyze and evaluate only the specific set of properties of the system that are under observation. Furthermore, simulation results are easier to analyze than experimental ones because important information at critical points can be easily logged to help researchers in diagnosing the network behavior. Starting from these premises, the proposed simulator architecture has been designed according to a fully modular scheme, to accommodate most of the control-plane specific characteristics of wavelength-routed network and provides a useful set of network traffic generators. The simulator models optical fibers and wavelengths individually for each link, thus allowing maximum flexibility in the representation of dissimilar networks. It also provides network utility subroutines to configure, monitor, and gather results and statistics about simulated networks. The resulting framework provides great configuration flexibility in topology definition (each node can or cannot support conversion capability; the number and type of lambdas, the associated maximum bandwidth and propagation delay can be specific for each link) and enables easy extension to introduce new features, not only in terms of RWA and grooming algorithms but also allowing other aspects of the simulator to be customized or expanded if desired in the future. The whole framework has been designed as an object-oriented application in which all functional entities are implemented by individual objects or modular entities interacting with each other. Each object can contain or refer to other objects for the composite identification of an operation. Objects can also be abstracted and encapsulated to facilitate extension. For example, if a new optical node type or new RWA algorithm in the simulator is needed, it can be

978-1-4244-4671-1/09/$25.00 ©2009 IEEE

conveniently added by extending the existing network device object or RWA algorithm object. The key components of the simulator can be divided into physical layer abstractions, such as optical switching devices and multi-wavelength fiber links, and logical layer modules working upon them, such as the RWA simulation engine and the traffic generation module, which together create and maintain the virtual topology.

3.1. Modeling the physical layer

The OXC object models the various types of switching nodes that constitute the Transport Plane of the simulated optical network. OXCs are responsible for switching traffic from an incoming fiber/wavelength pair (or electrical link) to an outgoing one. The necessary mapping information is maintained in a special internal table, called the switching matrix, which the OXC looks up before accomplishing each switching operation. According to their routing, traffic grooming and wavelength conversion capabilities of their interfaces, OXC objects can be used to model both hybrid optical routers, add and drop multiplexers or pure wavelength routers, operating in transparent or opaque mode. They provide wavelength routing & switching, wavelength conversion, fiber/port switching, waveband switching and regeneration. The simulated optical switching functions are supported in either transparent optical switching architecture or in the opaque one with O-E-O conversions. According to the underlying architecture the lambda conversion/switching capabilities may have different limitations, such as the number of converters in an OXC and the range of conversion. Each fiber features an independent WDM capability, i.e. consists of several wavelengths and the actual number of wavelengths is a parameter that can be configured for each fiber. Moreover, within each fiber strand connecting a couple of OXC objects there is one additional virtual link representing the dedicated communication channel needed to convey control plane traffic.

3.2. The RWA engine

The RWA simulation engine that is the heart of our framework is composed of two components: routing module and wavelength assignment module.

3.2.1. Routing. The routing module supports the dynamic creation and deletion of unidirectional lightpaths by determining the routes needed to establish such paths in the current network topology and with the available resources. It maintains two distinct views of the network – the physical and the virtual topology – represented as multi-graphs in which each edge corresponds to an individual channel (wavelengths), each fiber can support multiple channels, and there can be more than one fiber connecting the same pair of nodes.

The physical topology represents the real network infrastructure where every link between adjacent switching nodes and its resources (wavelengths and bandwidths) and conversion capabilities are catalogued. The virtual topology represents the current network status, dynamically modified during the creation of lightpaths. Multiplexing and demultiplexing wavelengths is simulated implicitly in the virtual topology forwarding plane. The simulator switching logic within the RWA engine will ideally bridge the incoming wavelength and outgoing wavelength to set up lightpaths. In a configurable manner, lightpath set-up is realized either first on existing available lightpaths (according to a best fit strategy in case there are several lightpaths satisfying the connection request), or directly executing the RWA algorithm without consulting the lightpath allocation tables. In the current implementation, the RWA engine already provides some basic predefined algorithms, such as Dijkstra shortest path algorithm/Minimum Hop Algorithm (MHA), Shortest Widest Path Algorithm (SWP), MINimum Residual Capacity algorithm (MinRC), k-shortest path algorithm and Minimum Interference Routing Algorithm (MIRA) [6][7][8].

3.2.2. Wavelength Assignment. The wavelength assignment module is responsible for selecting the wavelengths needed for lightpath creation. Our simulator supports optional wavelength conversion capability on each OXC. Also, channel allocation is done separately for each link. Up to 64 wavelengths can be used for transmission between network elements. More than one wavelength channel can be assigned between two network elements depending on traffic demands. Wavelength reuse is employed when possible. Most of the predefined algorithms of our RWA engine treats routing and wavelength assignment as a single joint problem, that is they search paths for connection requests considering both the routing and the wavelength assignment activities at the same time. Nevertheless, our simulator supports also disjoint RWA, in which the above activities are managed as two distinct problems. First the routing algorithm finds a suitable path for the connection in terms of available fibers between source and destination nodes; then, the wavelength assignment procedure provides wavelength assignments according to a specific criterion aimed at optimizing some resources. Different wavelength assignment algorithms have been implemented: first-fit, most-used/pack, least-used/spread, random allocation, round robin. With the first-fit algorithm, the first available wavelength is chosen, causing the wavelength with the smallest index to be chosen more often. The most-used/pack algorithm selects the available wavelength which is currently utilized on the largest number of fibers, thus preserving the maximum number of different wavelengths for future requests that may have the wavelength continuity constraint; analogously, the least-used/spread algorithm selects the available wavelength which is currently utilized on the smallest number of fibers, thus trying to balance the load on the network. The random rule distributes the connection traffic randomly so that average wavelength utilizations are balanced. In the round robin allocation scheme wavelengths are indexed and assigned in a circular manner, thus maximizing the use of all the available wavelengths in the network in order to minimize the blocking probability due to wavelengths unavailability. Finally, the bandwidth on a wavelength can be divided into smaller sub-rate capacities called sub-wavelength units, to be assigned to specific end-to-end connections. A connection request can demand one or more sub-wavelengths to a maximum of the whole available bandwidth supported on a specific wavelength, by performing traffic grooming from properly capable devices. Our simulation environment supports both single-hop and multi-hop grooming capability [9]. Single-hop grooming will be only possible on end-to-end connections realized on a single wavelength. On the other side, multi-hop grooming offers maximum flexibility in allowing sub-wavelength allocation across multiple chained lightpaths or single lightpaths built on multiple wavelengths (when the continuity constraint is relaxed). Clearly our network model allows the RWA algorithms to be aware of the different cost and complexity of the two mechanisms.

3.2.3. Signaling. The simulated control plane signaling logic is based on a reverse path reservation model that can be viewed as a simplified subset of RSVP-TE. The connection setup procedure is divided into two phases: downstream and upstream. During the downstream phase, each of the OXC nodes determines whether the required destination is reachable and there are enough resources to accommodate the new connection. As soon as the RWA engine determines on each node both the outgoing fiber and a wavelength with adequate bandwidth resources, it temporarily allocates them by adding a new record on the switching matrix for the new connection. At this instant, the setup is still incomplete, and therefore the switching matrix record is flagged as provisionally reserved and cannot yet be used by the OXC. At each step, the involved node passes the request to the downstream OXC object. If there is no fiber/wavelength acceptable for routing the current request or the available bandwidth is inadequate, the node rejects the requests and sends a response back to the source. The upstream phase starts as soon as the request reaches the egress node of the connection that acknowledges it along the same path. During this phase, when any node receives an acknowledge from its downstream node, it permanently allocates the resources reserved during the downstream phase, and modifies the switching matrix to activate the switching record for the new connection. As soon as the acknowledge is received by the ingress node, the connection is established and the new lightpath is modeled as a new direct edge in the virtual topology multi-graph, henceforward called cut-through edge (see [9]), with the capacity set to the difference between the minimum capacity on all the edges belonging to the lightpath and the fraction of the link bandwidth required by the involved connection request. A cut-through edge can be used in any path selection operation and thus can participate in one or more groomed paths as a single virtual edge (a single hop at the IP layer). When an established lightpath is torn down because the last connection occupying it is ended, the cut-through edge is removed and the edges in the extended graph corresponding to the underlying physical links are set back with full capacity. This schema allows to flexibly model nearly any network topology and to consider node conversion capabilities, wavelength availability and residual bandwidth per logical link at the IP layer. Each new connection request can be routed over a direct lightpath modeled as a single cut-through edge in our multi-graph, or over a sequence of lightpaths (a multi-hop path at the IP level, where each hop can be a lightpath), if it crosses lambda-edge or routers as well that will link together lightpaths.

3.3. The traffic generation module

For each given WDM network topology, the traffic generator module randomizes source and destination pairs according to a uniform distribution or a user-provided traffic matrix. Thus, traffic, expressed as the arrival of new connection request, may flow between any source–destination pairs. Along its path, each connection will require some amount of the sub-wavelength units according to its bandwidth requirement. If the requirement of a call cannot be fulfilled when it arrives, the request will be blocked. The probability that a new request will be blocked is, thus, an important indicator of the overall efficiency provided by networks. Simulation is a rather indispensable method to estimate the blocking probability, as well as other network performance parameters. In large optical networks, call blocking probabilities may be rare events due to the large capacity of the network or a very low request arrival rate. In such cases, standard simulation may require an extremely long runtime, and usually incurs in large relative errors. Connection requests may arrive or be torn down at any moment during the simulation. We developed a session traffic source object which generates three types of traffic, namely defined by a Poisson, exponential on/off and Pareto distribution with different average bit rate per wavelength. In particular, the Poisson distribution is a very good model for human requested end-to-end connections (i.e. semi-permanent virtual circuits) whereas the Pareto one is very useful for modeling typical Internet traffic-driven activity. In order to get more confidence in the simulation results, we provide the possibility to run multiple simulations on the same network with the same algorithm and parameters and eventually get the average result values.

4. Implementation details

The simulator architecture has been realized in modular form. This design choice stems from the nature of the problem and of the networks to be modeled. Each module is implemented as a collection of cooperating objects and each network component is represented by a separate object characterized by its own attributes and methods. A small number of parameters can be easily determined from simulation for nondestructive measurements. Objects and modules have been used as the building blocks of the whole simulation program. All objects have been designed in unified modeling language (UML), and implemented in Java. UML enabled to show the block diagram of the simulator as well as the simulation flow and the interaction among objects. Java has been chosen in order to take advantage of its object oriented paradigm, great extensibility, ease of modifiability, portability, and object serialization (to checkpoint out simulations). Besides, Java provides also the garbage collector that automatically frees the unused memory, thus simplifying the code and optimizing memory occupation. The simulator can run on workstations with the Java Virtual Machine, e.g. MS-windows or various UNIX machines. The simulator builds a model of the network in the multi-graph object, which is the main object representing the network topology; it is made up of the sets of nodes, fibers, and edges (initially each edge corresponds to a single wavelength). There is also the set of the paths currently routed in the network that grows up as the connection requests are honored. Besides there is a set of support objects build up and maintained for efficiency sake: these objects comprise indexes, vectors and matrices that have proven to speed up the computational time of the simulations. Each object has been designed keeping in mind efficiency, usability and flexibility. Abstract objects are used to uniform interfaces and are extended and instantiated as concrete objects representing real network elements. Object oriented encapsulation, inheritance and polymorphism have been exploited thanks also to the use of design patterns. The simulator also includes an intuitive graphic user interface (GUI) allowing flexible definition and modification of network topology and simulation parameters; a versatile configuration language is used to define complex simulation environments. The configuration file structure mainly consists of the physical topology information, such as nodes and links, and the source/destination enabled nodes and their properties in terms of bandwidth range, its distribution and related parameters. After execution of the simulation a detailed trace file is generated which shows the exact timing of the important events that have occurred during the course of simulation with the relevant details.

5. Functional evaluation

In order to validate the simulator, we ran many simulations based on well-known algorithms on both test network topologies and real networks such as NSFNet and Geant2, that have all been successfully modeled and tested. As performance measures, both throughput and blocking probability have been adopted. Figure 1 shows the blocking probability for NSFNet network topology with various known RWA algorithms.

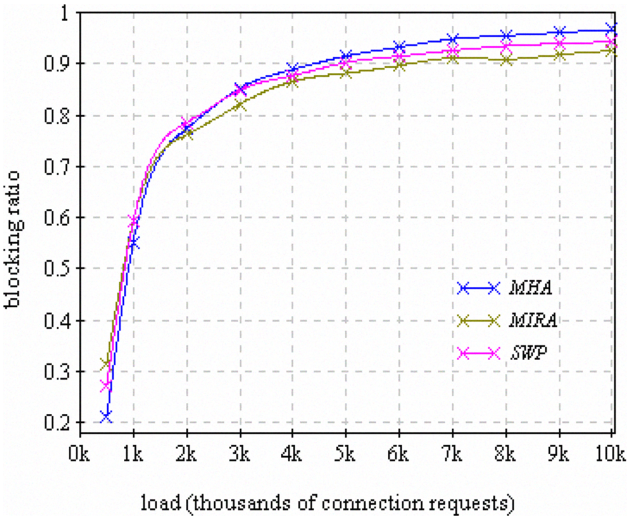

Figure 1. Average rejection ratio on NSFNet.

We can immediately observe from the above Fig. 1 the behavior, in terms of rejection ratio under increasing load, of some test RWA algorithms that closely matches several results available in literature (e.g. [10]). We also tested reliability of our simulator over variable network spacing from ring to full-mesh topology, as in [5] (see Fig. 2). Starting from a simple 8 nodes ring network with 8 fiber links and 32 wavelengths per link, new links are added stepwise up to reach 32 links. At each step, either the number of wavelengths on each link (series #1) or the wavelength transport capacity (series #2) is decreased in order to keep constant the network transport capacity, thus obtaining two series of test networks, as in Table 1.

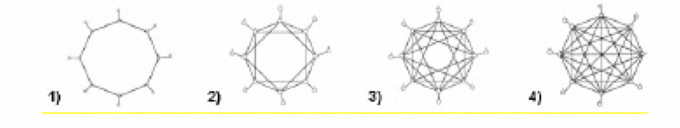

Figure 2. Ring to Full-Mesh test networks topologies.

Test series #1: λ varying, OC-unit constant				
Test network	***1***	***2***	***3***	***4***
Fiber links	8	16	24	32
λ per link	32	16	10	8
OC-unit per λ	24/48	24/48	24/48	24/48
Test series #2: λ constant, OC-unit varying				
Test network	***1***	***2***	***3***	***4***
Fiber links	8	16	24	32
λ per link	4	4	4	4
OC-unit per λ	768	384	256	192

Table 1. Test networks topologies parameters.

The Dijkstra shortest path routing has been used to route connection requests having a bandwidth requirement between 1 and 12 OC-units and uniform distribution between source/destination nodes is assumed. The results are shown in Fig. 3.

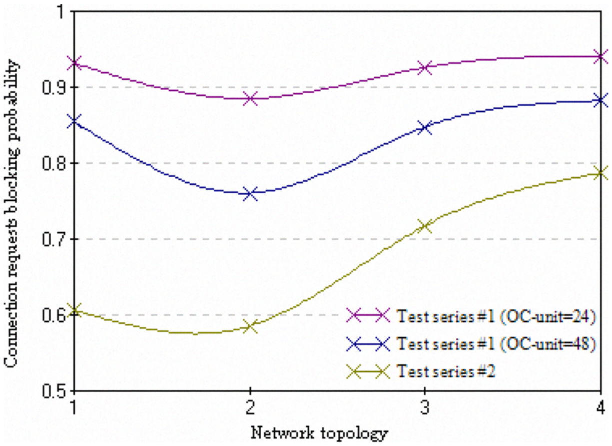

Figure 3. Simulations on Ring to Full-Mesh topologies.

As expected, better performances are achieved with a good balance between the number of channels (wavelengths) and transport capacity (OC-unit) on each channel. The first network is penalized because of the few links between node pairs, while the last because of either the small number of wavelengths per link (series #1) or the low transport capacity (series #2). Comparing the results of tests over known real networks and the effects of varying wavelengths and traffic load for different network topologies, we can conclude that our simulator is implemented properly and could be useful for optical network simulations.

6. Conclusion

Simulators have proven to be an indispensable tool for the study of the rapidly evolving optical networks technologies. In this paper we have presented SimulNet, a flexible simulator for WDM-routed networks. SimulNet has been expressly realized for the design and the evaluation of both RWA and optimization algorithms as well as for the validation of signaling protocols and scheduling schemes. SimulNet has shown good flexibility in managing even complex networks and has exhibited accuracy of simulations results and satisfactory performances, making it an useful tool for the optical network research community.

7. References

[1] Kim,Y., et al., "GLASS (GMPLS Lightwave Agile Switching Simulator) - A Scalable Discrete Event Network Simulator for GMPLS-based Optical Internet", *Proceedings of the JCCI'02* conference, 2002.

[2] Ogielski, A.T., SSFnet. Presentation at the *DARPA Next Generation Internet Conference*, Arlington, VA, 1999.

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

[4] The Network Simulator(ns), http://www.isi.edu/nsnam/ns/.

[5] Pedrola, O., Rumley, S., Klinkowski, M., Careglio, D., Gaumier, C., Solé-Pareta, J., "Flexible Simulators for OBS Network Architectures", *Proceedings of the IEEE ICTON 2008* conference, 2008.

[6] Guerin, R., Williams, D., and Orda, A., "QoS Routing Mechanisms and OSPF Extensions", *Proceedings of GLOBECOM*, 1997.

[7] Kar, K., Kodialam, M. and Lakshman, T. V., "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 921–940, 2000.

[8] Kar, K. Kodialam, M. and Lakshman, T. V., "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks", Proceedings of *IEEE Infocom* conference, *2001*.

[9] Balasubramanian, S., Somani, A. K., "On Traffic Grooming Choices for IP over WDM networks," Proceedings of *IEEE Broadnets* conference, 2006.

[10] Boutaba, R., Szeto, W., Iraqi, Y., "DORA: Efficient Routing for MPLS Traffic Engineering", J*ournal of Network and Systems Management*, Vol. 10, No. 3, 2002.

978-1-4244-4671-1/09/$25.00 ©2009 IEEE