JAVOBS: A Flexible Simulator for OBS Network Architectures

Oscar Pedrola, Miroslaw Klinkowski*, Davide Careglio, Josep Solé-Pareta
Universitat Politècnica de Catalunya (UPC), Jordi Girona 3, 08034 Barcelona, Catalunya, Spain
* and Instytut Łączności (IŁ), 1 Szachowa Street, 04-894 Warsaw, Poland
Email: {opedrola, mklinkow, careglio, pareta}@ac.upc.edu

Sébastien Rumley, Christian Gaumier
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
Email: {sebastien.rumley, christian.gaumier}@epfl.ch

***Abstract*— Since the OBS paradigm has become a potential candidate to cope with the needs of the future all optical networks, it has really caught the attention from both academia and industry worldwide. In this direction, OBS networks have been investigated under many different scenarios comprising numerous architectures and strategies. This heterogeneous context encouraged the development of various simulation tools. In this paper we present our novel Java-based OBS network simulator called JAVOBS. We discuss its architecture, study its performance and provide some exemplary results that point out its remarkable flexibility. This flexibility should permit an easy integration of upcoming new network protocol designs but also support changing and evolving research goals.**

***Index Terms*— Optical burst switching (OBS), simulation tool, flexibility, performance evaluation.**

I. Introduction

To move towards IP-over-WDM architectures, various optical switching techniques have been under intensive research. Among them, three switching paradigms appeared as potential candidates. First, Optical Circuit Switching (OCS) [1] pursues a wavelength routed networking architecture with a whole wavelength as finest granularity. However, it lacks both the flexibility and efficiency required to cope with the needs of current traffic patterns. Second, in the Optical Packet Switching (OPS) approach [2], each packet is sent into the network together with its own header. This header is either electronically (or even roughly all-optically [3]) processed at each intermediate node while the packet is optically buffered. Although OPS may be seen as both the natural choice and conceptually ideal for the future all-optical networks, current optical technology is still immature and not able to overcome its exigencies. Finally, in order to provide optical switching for next-generation Internet traffic in a flexible yet feasible way, the Optical Burst Switching (OBS) paradigm was proposed in [4][5]. In OBS networks, burst control packets (BCPs) are sent out-of-band both to reserve all resources and to set up the path for their associated data bursts, which will be sent optically after an offset time in a cut through manner. In this way, OBS allows for an efficient use of resources without the need of optical buffering at any intermediate node. Although it can be seen as an intermediate step of the migration from OCS to OPS, OBS has emerged as a more competitive choice for the transmission of data traffic in the near future. In essence, OBS combines the best from both OCS and OPS while avoiding their shortcomings. Consequently, OBS has received an increasing amount of attention from the optical research community and has become, nowadays, a research field of its own.

OBS networks display a complex structure and the design of their constituent elements offers several degrees of freedom. So far, much of the research on OBS networks has been conducted through theoretical analysis. Undeniably, the analytical approach can provide valuable insights in reduced complexity scenarios but might scarcely cope with the multiple factors that hide behind a complete network schema. Simulation tools have become essentials to evaluate complex OBS network scenarios. Indeed, simulators solve many difficulties such as the need to build a real system, but more important, they allow for the reproducibility of results, which is the basis for scientific advance [25].

In this paper, we present our Java-based OBS network simulator (JAVOBS), which was firstly presented in [24]. Considering how rapidly new strategies are engineered to improve the performance of OBS networks, it is our objective to demonstrate how versatile a simulation tool should be in order to be able to provide reliable results in a relatively fast yet straightforward way. Section II gives an insight of the wide variety of OBS schemes proposed so far as well as it reviews OBS network simulation tools presented in the literature. Section III presents the architecture of the JAVOBS simulator. Section IV provides some numerical results that both validate the simulator and show its flexibility when implementing different OBS protocols and algorithms. Section V summarizes the flexibility and extendibility of JAVOBS. We conclude this paper in Section VI.

doi:10.4304/jnw.5.2.256-264

II. OBS REVIEW

An OBS network is made up of two types of nodes, namely edge and core nodes. Edge nodes are in charge of both assembling input packets coming from different sources (e.g. IP, Ethernet) into outgoing bursts and of disassembling incoming bursts. For each outgoing burst, edge nodes emit a separate BCP in advance, to reserve resources (i.e. bandwidth on a desired output channel) along the way from the ingress node to an egress node. Core nodes in OBS are responsible for switching individual bursts and for reading, processing, and updating burst control packets. Core nodes are generally assumed wavelength conversion capable.

The BCP carries, among other information, the remaining offset time at the next hop (i.e. the time separating the arrival of the BCP from the arrival of the burst), and the burst length.

A. Burst Reservation Protocols

In order to transmit bursts over an OBS network, a resource reservation protocol must be put in place to ensure the allocation of resources and to properly configure the optical switch before the corresponding data burst arrives at the node. Two different approaches were designed. A wavelength-routed OBS reservation protocol was proposed in [7] as a two-way reservation scheme (i.e. a burst cannot be sent without the successful reception of an acknowledgement). Nevertheless, much of the research has been devoted to the one-way reservation scheme aiming to reduce the light-path setup time and consequently increase the resource utilization in OBS networks. The just-in-time (JIT) [8], Horizon [5] and just-enough-time (JET) [4] resource reservation protocols are the most well-known one-way reservation schemes. More recently, JIT+ [9] and E-JIT [10] protocols have also been proposed. The main difference between all one-way reservation schemes stems from the manner in which output wavelengths (i.e. channels assuming wavelength conversion) are reserved for bursts. These schemes include: (a) immediate reservation (JIT, E-JIT); (b) delayed reservation with void filling (JET); (c) delayed reservation without void filling (Horizon); (d) modified immediate reservation (JIT+).

A comparison of the JIT, JIT+, JET and Horizon protocols can be found in [9]. Delayed schemes produce a more efficient use of resources, especially when void filling is applied, and perform better in terms of burst loss probability. However, the sophisticated scheduling algorithms that they require increase the processing times of BCPs at intermediate nodes. Thus, in such scenario, the simplicity of JIT may balance its relative poor performance [9]. Indeed, in contrast to the other protocols, hardware implementations of the JIT signaling protocol have already been realized and published [11].

B. Burst Scheduling

When a core node receives a BCP, it must decide which output channel should be reserved to later forward the burst corresponding to this BCP. Scheduling algorithms aim to transfer efficiently the input traffic to the desired output while configuring the switching matrix adequately.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT SCHEDULING ALGORITHMS

Scheduling Algorithm	Time Complexity	Bandwidth Utilization
FFUC	O($log\ w$)	Low
Horizon / LAUC	O(w)	Low
LAUC-VF	O($w\ log\ N_b$)	High
FFUC-VF	O($w\ log\ N_b$)	High
Min-SV/EV	O($log^2 N_b$)	High

To date, several algorithms have been proposed to solve the wavelength scheduling problem in OBS networks. They can be divided into two sets depending whether they perform void filling (a) or not (b). Algorithms belonging to group (b) pursue simplicity when searching an available wavelength. They are not aimed to maximize the use of resources but to generate low processing times. A simple scheduling algorithm based on the Horizon reservation protocol and called latest available unused channel (LAUC) , was proposed in [8]. Another example is the first fit unscheduled channel (FFUC) algorithm [13].

More advanced scheduling algorithms belong to group (a). These algorithms are designed both to provide efficient use of resources and to reduce blocking probabilities. However, void filling algorithms are more complex, hence difficult to implement and imply high processing times. Among the void filling algorithms one finds: (1) latest available unused channel with void filling (LAUC-VF) [12]; (2) first fit unscheduled channel with void filling (FFUC-VF) [12]. More recently, the minimum starting void (Min-SV) and the minimum ending void (Min-EV) scheduling algorithms were presented in [14]. Min-SV and Min-EV algorithms improve significantly the processing time over LAUC-VF. However, Min-SV/EV algorithms involve time-consuming memory accesses. Therefore, the void filling algorithms are still considered too slow to provide a viable solution to the problem [15]. Table 1 summarizes the comparison between the algorithms based on the study in [16]. It uses the following notation: (w) number of wavelengths at each output port; (N_b) number of bursts currently scheduled on every wavelength.

C. OBS Simulation Tools

OBS networks are still in a phase where several options may have their own opportunity. Therefore, there is a strong need to mimic the behavior of real OBS networks. That is precisely the task of simulation tools. Since OBS is a relatively young field, much of the studies that can be found in the literature use quite simple simulation models. For instance, several proposals have been applied only to a single node [9][21]. In general, these simulation models were developed in purpose for a specific situation and are not suitable to study complete

OBS scenarios. On the other hand, some well-known simulators such as the widely known ns-2 [17] or the IKR Simulation Library [18], have or allow extensions for the study of OBS networks. A comparison of some existent OBS simulator tools can be found in [19]. To our best knowledge, none of them was specifically developed for the study of OBS networks, and thus do not provide support to the full set of OBS representations. Besides, given their divergence of perception of the OBS scenario it is not possible to compare their results [19].

In consequence, other tools exclusively aimed to analyze OBS networks have been proposed. Two new simulation models are presented in [20][21]. Both exploit the object-oriented approach using either the C++ in the former case or the Java programming language in the latter. The common goal of these new models is to reach the flexibility degree that simulation of OBS networks requires. Following a modular construction process, a high degree of flexibility is exhibited. At the same time, the introduction of further developments is facilitated.

Yet another OBS network simulator (ADOBS) has been developed in C++ [6]. Formerly, ADOBS served to study routing algorithms in OPS networks. Lately, it has been modified to become an ad-hoc event-driven simulator for OBS networks. ADOBS has been basically used to study the performance of the OBS network layer. Since C++ is a low level programming language, the developer deals with concepts and operations strictly connected with computer hardware. Hence, speed and efficiency are achieved at the cost of complexity.

III. JAVOBS Architecture And Features

The JAVOBS simulator is a Java-based application that has been exclusively built to simulate OBS networks on top of the JAVANCO framework [22].

A. The JAVANCO Framework

The JAVANCO framework is programmed within the Java 1.6.0 platform, using the popular Java programming language. It has been conceived to provide a coherent object oriented structure that is able to properly represent graph and network topologies in a compelling yet versatile way. Over this fundamental structure, several packages offer a variety of features including graphical visualization, support for disk serialization of topologies and execution of common graph algorithms. It is thanks to these core packages that the user can rapidly develop and test network planning procedures through the construction of simulation models.

By its nature, Java is an interpreted language. This means that user code is temporarily compiled into "Java byte code", and does not become executable code until the program is actually run. Consequently, C++ runtime performance is better than that of Java. Nevertheless, Java has been selected both to avoid the complexity of building a simulator completely from scratch using C++ and to benefit from the many advantages provided by the Java environment. In particular, Java being a garbage-collected language, the procedures of memory handling are greatly simplified.

Figure 1 shows a general representation depicting the architecture of JAVANCO. The cornerstone of its architecture is the *NetworkHandler* object, in charge of both organizing the references towards each object composing the graph (i.e. layers, links and nodes) and providing access to several managers and engines (e.g. user interface manager, serialization manager, script engine). JAVANCO permits to load and save files that describe network topologies and their components. This functionality makes use of the Multilayer Network Description (MND) proposed in [23], which is based on the XML standard. Taking advantage of this description format, it is easy to associate several attributes to any element present in a network topology (e.g. the capacity in a link).

JAVANCO embeds a script engine which allows calling any functionality of the framework and dispenses the user to write complete Java classes. It also supports different kinds of user interfaces.

B. JAVOBS Features

Two general models exist to conduct discrete simulations: next-event time progression and fixed-

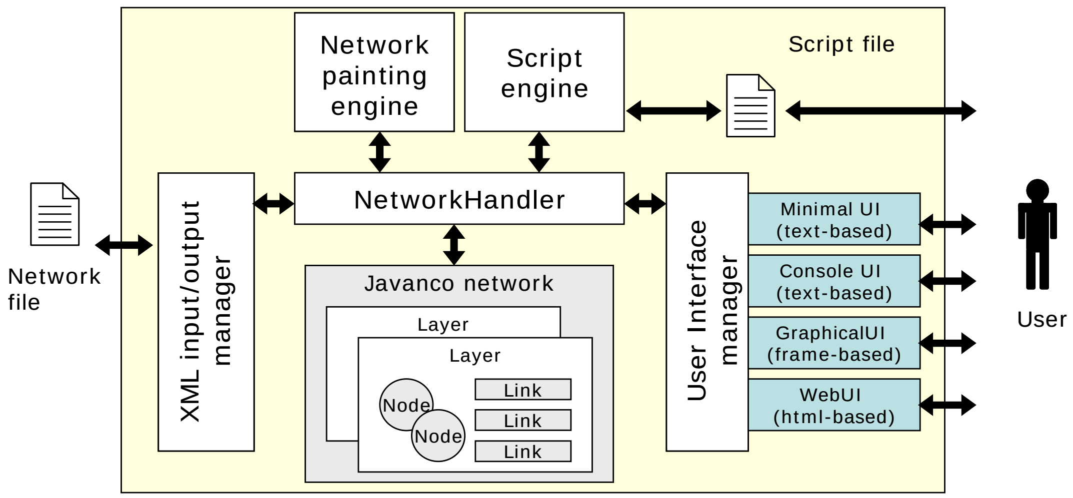

Figure 1. General view of the JAVANCO Architecture.

increment time progression [21]. The JAVOBS core simulation engine implements both. Each simulation is thus separated in fixed length steps, but events can occur within a step with fine time granularity. Arrivals of bursts at either intermediate or egress nodes are implemented with events. Other activities such as traffic generation and reservation list updates are executed at the beginning or at the end of each step. This hybrid simulation scheme mainly permits to generate the traffic progressively and thus keep the memory usage moderate. Furthermore, it also permits, under a minor constraint, to execute concurrently all the events expiring in one step, and therefore to take advantage of the parallel computing functionalities provided by recent CPUs. The constraint is not restrictive at all. Only the step time length should be shorter than the propagation delay of the shortest link.

JAVOBS offers a basic OBS model which can be adapted and modified in many ways. Using the functionality offered by JAVANCO, for example multiple OBS network topologies can be constructed over various *NetworkHandler* objects. These topologies can either be dynamically constructed, graphically designed, or loaded from MND files. Also, many of the aforementioned reservation protocols and scheduling algorithms have been implemented within JAVOBS to enhance the basic OBS model.

It has to be mentioned that the JAVOBS model offers a shortest path routing scheme, which uses link lengths defined in the JAVANCO *NetworkHandler* to compute shortest paths. This minimal routing logic can however be replaced by a more sophisticated one. JAVOBS offers two options to define the routing logic: (a) one unique routing element is defined, this element being then responsible for all routing operation during the whole simulation; (b) one routing element per node, each one supporting independent configuration.

TABLE II.
ADOBS/ JAVOBS FEATURES COMPARISON

Feature	ADOBS	JAVOBS
OBS Protocols	JET, Horizon	JET, JIT, Horizon, E-JIT, JIT+
Scheduling Algorithms	LAUC, LAUC-VF FFUC, FFUC-VF	LAUC, LAUC-VF FFUC, FFUC-VF
OBS Architectures	C-OBS, E-OBS	C-OBS, E-OBS
Model Building	Predefined input file	Graphically edited, input file, or created dynamically
Routing	Specified	Configurable
Traffic characterization	Fixed (Poisson)	Configurable
Programming Language	C++	Java

JAVOBS offers another degree of flexibility with respect to the traffic generation. In many case studies, traffic characteristics are supposed to be independent of the source or destination node. In these cases, a unique traffic generator can be used to generate all burst sizes and departure times. However, JAVOBS also allows equipping each edge node with an independent burst generator, permitting in this way studies involving source and/or destination dependent traffic flows.

Eventually, JAVOBS natively supports simulations of the emulated offset time control architecture (E-OBS) [4] along with the conventional OBS control architecture (C-OBS). In Table 2 we summarize the abilities of JAVOBS and compare them with the aforementioned ADOBS simulator.

IV. SIMULATION RESULTS

The purpose of this section is to evaluate the performance and the flexibility of the JAVOBS simulator. We first provide results related to simulator validation and runtime tests. Second, we focus on four different case-studies: (1) Performance comparison of reservation protocols under both the C-OBS and the E-OBS control architectures supported in JAVOBS; (2) Comparison of the Horizon and the Constant Time Burst Resequencing (CTBR) [15] schedulers under the single node topology; (3) Analysis of the network topology flexibility using different degrees of meshed-rings; (4) Evaluation of the network-wide burst loss performance with different burst traffic statistics. Apart from presenting performance results we take the opportunity to discuss some OBS specific issues and, in particular, compare different OBS architecture and protocol proposals.

A. Validation and Benchmarking of JAVOBS

In order to assess its credibility, the JAVOBS simulator has been validated by means of an analytical model for the calculation of the network-wide burst loss probability and by comparison with results obtained with the ADOBS simulator. The analytical results are based on a reduced link load model for OBS networks presented in [26].

We use within both ADOBS and JAVOBS simulators a network topology called SIMPLE [6] with 6 nodes and 8 links, and compute an identical shortest path routing. Each node is an edge node generating 25.6 Erlangs (0.8, when normalized to the link capacity) and each link has a capacity of 32 channels. Bursts have exponential distributed arrival time and length. To keep relation with the real world, we set the channel capacity to 10 Gbit/s and the burst mean size to 1 Mb. In obtaining the simulation results, we estimated 99% confidence intervals. Since the confidence intervals found are very narrow, we do not plot them in order to improve readability. As it can be seen from Figure 2, the results obtained by JAVOBS match both the analytical results and the results obtained with ADOBS. Hence, in this case, we consider the simulator validated.

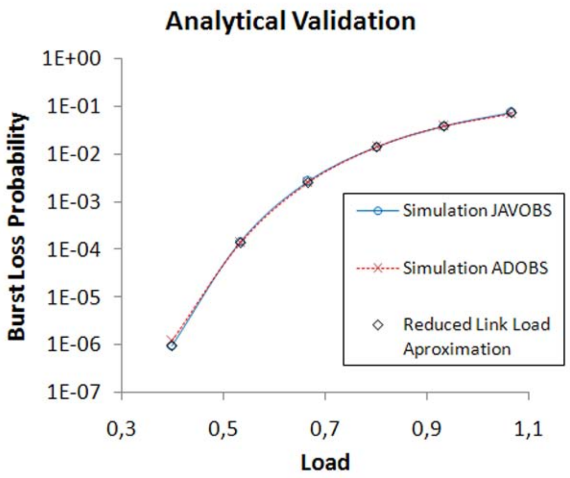

Figure 2. JAVOBS / ADOBS Analytical Validation.

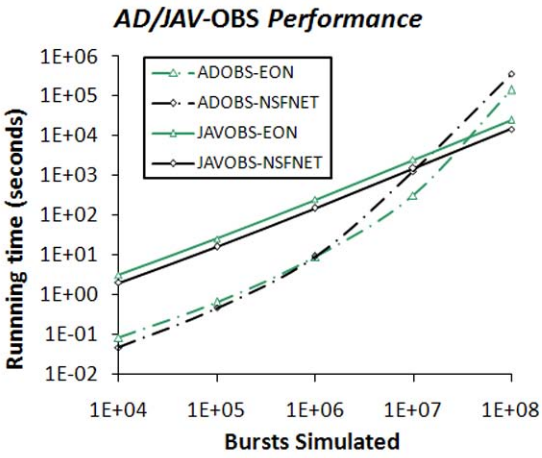

Figure 3. JAVOBS / ADOBS Runtime Test.

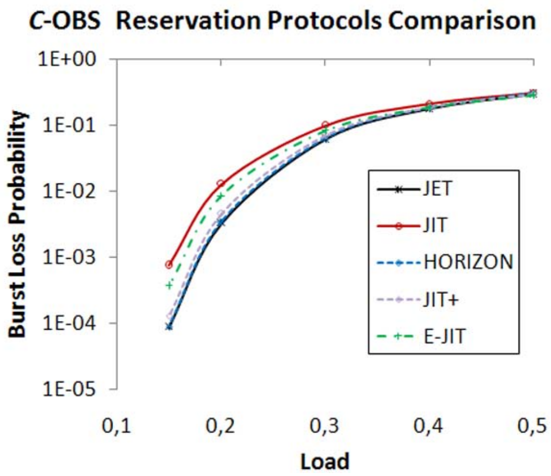

Figure 4. Burst Loss Probability vs. Load in C-OBS.

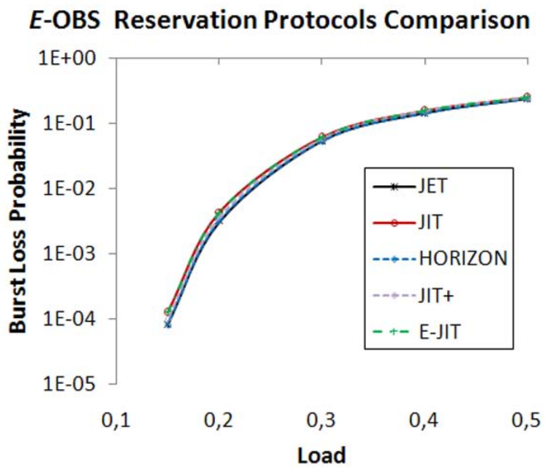

Figure 5. Burst Loss Probability vs. Load in E-OBS.

A test measuring the running time of both simulators has also been performed. Simulations were run according to the number of bursts generated and prompted more than one hundred hours of simulation (on an Intel Core 2 Quad 2.4 GHz desktop computer). In this case, we consider two network topologies: (1) NSFNET [27] (US network); (2) EON [28] (a pan-European network defined in European COST 266 action) with 15 and 28 nodes, and 23 and 39 links respectively. JET signaling and LAUC-VF scheduling are used.

The results obtained are shown in Figure 3. ADOBS performs better at low values of generated bursts, probably taking advantage of the C++ performance. However, tendency changes at about 1 million bursts. From this point on, the ADOBS curves exhibit an exponential increase which finally creates gaps of up to 96 hours between both simulators. This gap is apparently due to unoptimized memory utilization in the ADOBS simulator which obliges the operating system to use the hard disk as RAM extension, and thus, drastically reduces the simulator throughput. Although JAVOBS is outperformed in short simulations, we observe a constant growth of the running times for all time scales which exhibits its robustness. Thus, in this case, the benefits of using a garbage-collected language, which dispenses the user to take care of many memory management related operations, become apparent.

B. Evaluation of the E-OBS and C-OBS Architectures.

Considering that fiber delay line (FDL) buffers are not used, it has been proved in [29] that the best worst-case performance of an online best-effort scheduling algorithm is achieved when all bursts have the same offset time and the same length. One of the benefits of E-OBS comes from the fact that offset times are introduced at each core node by means of additional fiber delay coils inserted in the data path at the input port of the node. As a result, E-OBS does not experience offset variation inside the network. In such scenario, scheduling algorithms do not need to implement any void filling technique. Therefore, in an E-OBS network, JIT and Horizon reservation mechanisms seem to be the most appropriate ones due to its low complexity compared to JET. Indeed, the over-provisioning of resources that characterizes JIT is substantially reduced using E-OBS due to smaller offset times. Figures 4 and 5 present the results obtained in both control architectures under the different signaling protocols supported by JAVOBS.

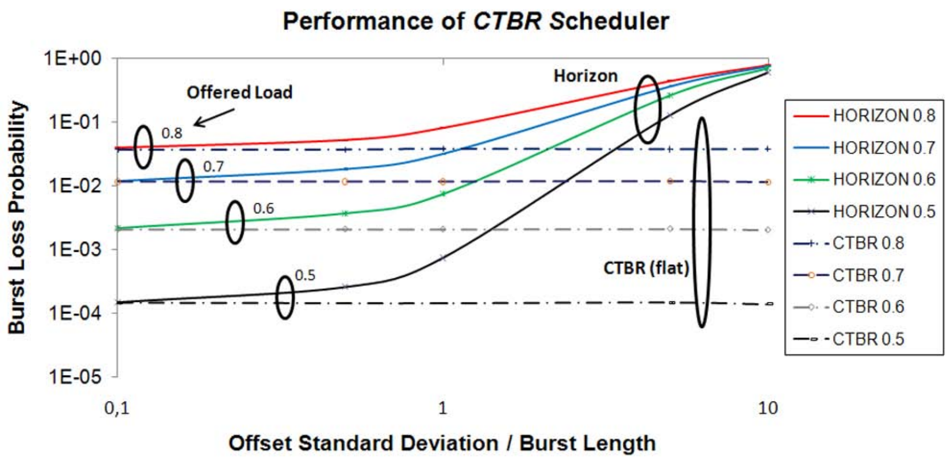

Figure 6. Performance Comparison of the Horizon and CTBR Schedulers under the Single Node topology.

We consider the EON network topology and a mean burst length of 40 kB. The processing and switching times are estimated according to [9] and [6]. We observe that using E-OBS, the performance of the five different signaling protocols is very similar, thus, the possibility of reducing the network complexity by using low complexity techniques such as JIT is not unfounded. On the contrary, in C-OBS becomes clear the advantage of using complex reservation mechanisms due to the variable offsets.

C. Implementation of the CTBR Algorithm.

Since OBS has ultra high speed requirements, the bandwidth efficient scheduling algorithms proposed so far are not considered a viable solution to the problem due to their large processing times. Recently, in [15], a hardware implementation of an optimal wavelength scheduler that can produce burst schedules in a time complexity of O(1) was presented. The idea consists of producing schedules by bursts arrivals rather than BCPs arrivals. The optimal wavelength scheduler consists of two components: (a) the CTBR block; (b) the horizon scheduler. It is important to notice that the driving force behind this technique is the simplicity of horizon and its ability to operate at high speed.

We developed a set of classes implementing this alternative OBS model. To perform the simulation, we used the parameters specified in [15] with the aim of comparing the results obtained. Since the topology utilized for the simulation is not mentioned, we assumed the single node implementation. The performance of both the Horizon and CTBR scheduler is compared. The offset times of all bursts are generated according to a lognormal distribution with mean 100µs. Figure 6 shows the results obtained. We observe a clear match with the results presented. The burst loss probability of the horizon scheduler increases when the ratio between the offset time standard deviation and the burst length increases. On the other hand, in the CTBR scheduler, the curves remain flat regardless of the ratio variation.

D. Flexible topology simulations.

The flexibility of JAVOBS has been tested in respect of the topologies with the aim of demonstrating that JAVOBS allows topological modifications in a straightforward way. We evaluate the simulator adaptability performing a set of simulations over different degrees of meshed-ring topologies.

The study begins with an 8 node ring topology with 32 wavelengths per link and ends with 28 links (full-mesh) and 9 wavelengths per link. At each step of the study, a bench of simulations is conducted on the topology. Then, topology is extended with additional links. However, in order to keep constant the network capacity, the number of wavelengths per link is recomputed at each step.

Figure 7 shows the results of the simulations conducted on each intermediate topology. A shortest-path routing algorithm has been used. The BCP arrival rate λ of BCPs is maintained constant for all scenarios. As expected, the blocking probability is evidently reduced as more direct links between each source-destination pair become available.

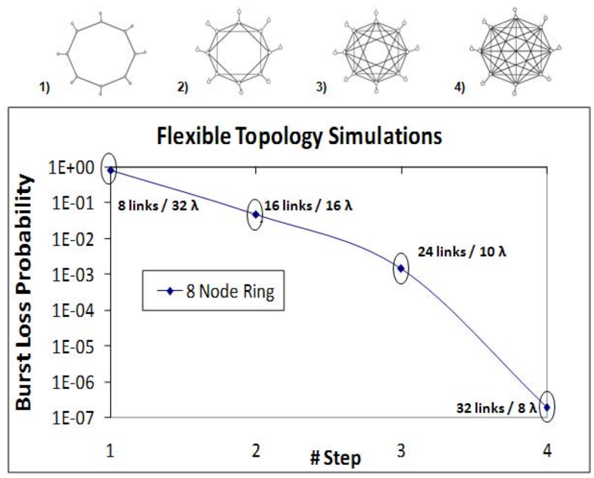

Figure 7. Ring Topology Study

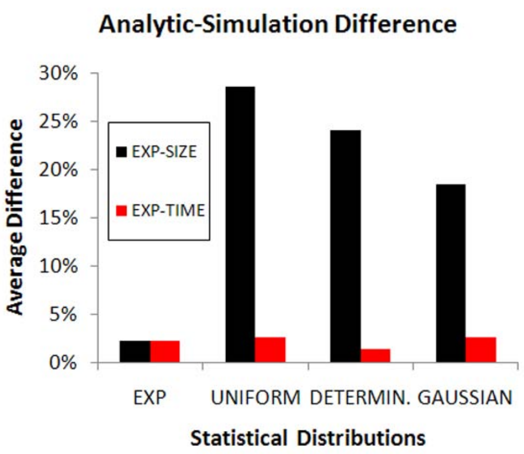

Figure 8. Difference between Analytical and Simulation results.

D. Impact of Burst Traffic Statistics on Burst Loss Performance .

Eventually, in order to study the network-wide burst loss performance, we used the JAVOBS capability of synthesizing diverse burst traffic statistics. Several statistical distributions are available for creating either burst arrival times or lengths, and consequently, different combinations of burst traffic statistics.

The analytical model that we used for the validation of our simulator does not depend on the burst length distribution chosen; however, it only holds for burst arrivals following a Poisson process. We thus evaluated the difference between the analytical model and the simulation results when burst arrivals follow statistical distributions different from the abovementioned Poisson. Concretely, we run two different sets of simulations.

In the first case, we preset the burst length distribution to *Exponential* with mean burst length equal to 1 Mb, and alternatively, we used *Exponential*, *Uniform*, *Gaussian* and *Deterministic* burst arrival distributions with their corresponding parameters set according to the load generated. Afterwards, we performed a second set of simulations; however, this time, we preset the burst arrival distribution to *Exponential*, and alternatively, the burst length distribution is replaced. In this experiment we used the NSFNET network topology and JET signaling together with LAUC-VF scheduling.

The results shown in Figure 8 agree with the analytical model. While the second set of simulations exhibits negligible differences with respect to the analytical values, the first set of results presents gaps of up to 28 % between both. Since the analytical model assumes exponential burst arrival times to compute the burst loss probability, these results verify, again, the right performance of the JAVOBS simulator.

V. SUMMARIZING JAVOBS FLEXIBILITY

The JAVOBS simulator we presented all through this contribution provides the user with a dedicated OBS simulation framework. This can be adapted, modified or extended in many ways, implying different level of confidence with the Java programming language and with the JAVOBS API. This section recapitulates what can be changed and adapted referring to the examples presented in the previous section.

Very fundamental parameters such as switching time or offered traffic rate are given at the beginning as input values. Results displayed on Figure 2 are thus straightforward to reproduce. To configure the reservation protocol and the scheduling algorithm is also a straightforward operation, which makes Figures 4 and 5 easily reproducible, too.

Changing the routing logic and the traffic generators (Figure 7) is slightly more complex. Basic knowledge of the Java language is required. Similarly, studies involving topological (Figure 8) modification at run-time require knowledge of the JAVANCO API and of Java.

To setup studies where traffic is generated according to specific rules (e.g. constant flows, self similar traffic or aggregation of finer granularity traffic), additional implementations of burst generators are required. In the same way, prototyping of more complex scheduling algorithms is also possible by implementing new classes. Additional classes have to comply with well defined rules and implement strictly defined methods.

Eventually, several changes have been required in the core simulation engine to generate the results of Figure 6 (CTBR). However, in obtaining these results, most of the functionalities developed for conventional OBS were reused, and thus, they did not involve a whole reimplementation of the simulator.

Table III summarizes the configurable or extendable parts of JAVOBS.

TABLE III
SUMMARY OF JAVOBS EXTENDIBILITY

Parameter or functionality	Requirements	Difficulty
Processing time, Switching time, Mean burst size, Offered rate, Simulation time, Number of simulations	none	Very low
Scheduling Algorithms, Reservation Protocols	OBS fundamentals	Very low
Topology (from file)	XML and MND knowledge	Low
Topology (dynamic modification)	Basic Java and Javanco knowledge	Low
Routing logic, Traffic generators	Basic Java knowledge	Low
Alternative scheduling algorithms, reservation protocols, traffic generator or routing logics	Java knowledge	Moderate
Alternative simulation schemes or OBS models	JAVOBS knowledge, Java knowledge	High

VI. Conclusions

We have presented our novel Java-based simulation tool JAVOBS, which has been exclusively developed for the study of OBS networks. We have also given a recent overview of the existent simulation tools for OBS networks. We have verified that comparisons between simulators were impossible due to their heterogeneity. The JAVOBS simulator has been described, validated and compared with an ad hoc C++ based simulator.

The flexibility of our simulator has been highlighted through a series of experiments that exhibit its performance. From the results of these experiments, it is concluded that: (1) as OBS networks are still undergoing intense research and development, its study requires simulation tools that facilitate the introduction of enhancements and new techniques, (2) as long as the simulation model is valid, flexible simulation tools such as JAVOBS can save time and computational resources.

Acknowledgment

The work described in this paper was carried out with the support of the CATARO-project (TEC2005-08051-C03-01) funded by the Spanish Ministry of Education and Science (MEC) and within the COST Action 291, supported by Swiss National Science Foundation.

References

[1] I. Chlamtac, A. Ganz, G. Karmi: Lightpath communications: An approach to high-bandwidth optical wans, *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171-1182, July 1992.

[2] D. Chiaroni: A novel photonic architecture for high capacity atm switch applications, *in Proc. PS 1995*, Salt Lake City, UT, April 1995.

[3] F. Ramos et al., “Ist-lasagne: towards all-optical label swapping employing optical logic gates and optical flip-flops”, IEEE/OSA J. Lightwave Technol., vol. 23, no. 10, Oct. 2005.

[4] C. Qiao, M. Yoo: Optical burst switching (OBS)-A new paradigm for an optical Internet, *J. High Speed Networks*, vol. 8, no. 1, pp. 69-84, Jan. 1999.

[5] J. S. Turner: Terabit burst switching, *J. High Speed Networks*, vol. 8, no. 1, pp. 3-16, Jan. 1999.

[6] M. Klinkowski: Offset time-emulated architecture for optical burst switching - modelling and performance evaluation, *Phd’s Thesis*, DAC department, Technical University of Catalonia (UPC), Feb. 2008.

[7] M. Duser, P. Bayvel: Analysis of a dynamically wavelength-routed, optical burst switched network architecture, *IEEE/OSA J. Lightwave Technol*, vol. 20, no. 4, pp. 574-585, April 2002.

[8] J. Y. Wei, R. I. McFarland: Just-in-time signaling for WDM optical burst switching networks, *J. Lightwave Technol.*, vol. 18, no. 12, pp. 2019-2037, Dec. 2000.

[9] J. Teng, G. N. Rouskas: A detailed analysis and performance comparison of wavelength reservation schemes for optical burst switched networks, *Photonic Network Commun.*, vol. 9, pp. 311-335, May 2005.

[10] J. J. P. C. Rodrigues, *et al.*: Enhanced Just-in-Time: A new resource reservation protocol for optical burst switching networks, *in Proc ISCC 2007*, Aveiro, Portugal, July 2007.

[11] I. Baldine, M. Cassada, A. Bragg, G. Karmous-Edwards, D. Stevenson: Just-in-time optical burst switching implementation in the ATDnet all-optical networking testbed, *in Proc. Globecom 2003*, San Francisco, CA, Dec. 2003.

[12] Y. Xiong, M. Vandenhoute, H. Cankaya: Control architecture in optical burst switched WDM networks, *IEEE J. Select. Areas Commun.*, vol.18, pp. 1838-51, Oct 2000.

[13] K. Dozer, C. Gauger, J. Spath, S. Bodamer: Evaluation of reservation mechanisms for optical burst switching, *AEU Internat. J. Electron. Commun*, vol. 55, no. 1, Jan. 2001.

[14] J. Xu, C. Qiao, J. Li, G. Xu: Efficient channel scheduling in optical burst switched networks, *in Proc. IEEE Infocom 2003*, vol. 3, pp. 2268–2278, San Francisco, CA, March 2003.

[15] Y. Chen, J. S. Turner, P.-F. Mo: Optimal burst scheduling in optical burst switched networks, *J. Lightwave Technol.*, vol. 25, no. 8, August 2007.

[16] V. M. Vokkarane, J.P. Jue: Segmentation-based nonpreemptive channel scheduling algorithms for optical burst-switched networks, *J. Lightwave Technol.*, vol. 23, no. 10, Oct. 2005.

[17] Optical Internet Research Center. OIRC OBS-ns simulator: http://wine.icu.ac.kr/~obsns/index.php, accessed at January 2009.

[18] Institute of Communication Networks and Computer Engineering (IKR), IKR Simulation Library: http://www.ikr.uni-stuttgart.de/INDSimLib/, accessed at January 2009.

[19] V. Soares, *et al.*: OBS simulation tools: A comparative study, *in Proc. ICC 2008*, Beijing, China, May 2008.

[20] M.Casoni, *et al.*: M_OBS_SIM: A powerful modular optical burst switched (OBS) network simulator, *Simulation Modelling Practice and Theory*, Elsevier Journal, available on line.

[21] J. Rodrigues, *et al.*: Object-oriented modelling and simulation of optical burst switching networks, *in Proc. GTC Workshops*, collocated with Globecom 2004, Dallas, TX, Nov. 2004.

[22] S. Rumley, C. Gaumier et al.: Software tools and methods for research and education on optical network, *in COST action 291 final report*, in press.

[23] S. Rumley, C. Gaumier: Multilayer description of large scale communication networks, *Recent Advances in Modeling and Simulation Tools for Communication Networks and Services*, pp.121-135, Springer, 2008.

[24] O. Pedrola, S. Rumley, M. Klinkowski, D. Careglio, C. Gaumier and J. Solé-Pareta: Flexible simulators for OBS network architectures, *in Proc. ICTON 2008*, Athens, Greece, June 2008.

[25] C. Phillips: A review of high performance simulation tools and modeling concepts, *Recent Advances in Modeling and Simulation Tools for Communication Networks and Services*, pp. 29-48, Springer, 2008.

[26] Z. Rosberg, Hai Le Vu, M. Zukerman, J.White: Performance analyses of optical burst-switching networks, *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, pp. 1187-1197, Sept. 2003.

[27] Nsfnet-the national science foundation network: http://moat.nlanr.net/, accessed at March, 2008.

[28] B. Mikac R. Inkret, A. Kuchar: Advanced infrastructure for photonic networks, *extended final report of Cost action 266*. Technical Report ISBN 953-184-064-4.

[29] J. Li, *et al.*: Maximizing throughput for optical burst switching networks, *in Proc. Infocom 2004*, Hong Kong, March 2004.

Oscar Pedrola received the M.S. degree in telecommunications engineering and the M.S. degree in information and communication technologies both from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain in 2008. Currently, he is a PhD student at the UPC, in the Broadband Communications Research Group (CBA). At present, he is involved in the FP7 Network of Excellence BONE. His current research interests are in the field of optical networks with emphasis on burst/packet based switching technologies.

Sébastien Rumley received M.S. degree in telecommunications engineering from the Swiss Federal Institute of Technology of Lausanne (EPFL) in 2005, after undergraduate studies in Lausanne and Zurich (ETH), and graduate studies in Lausanne and Santiago de Chile (PUC). Since 2006, he is working as research assistant at the Telecommunications Laboratory of the EPFL. His research focuses on mechanisms improving the collaborations between geographically distributed research institutes, in the context of COST actions 285 and 291.

Miroslaw Klinkowski is an Assistant Professor in the Department of Transmission and Optical Technology at the National Institute of Telecommunications in Warsaw, Poland, and a Postdoc at the Universitat Politècnica de Catalunya (UPC) in Barcelona, Spain. He received his M.Sc. and Ph.D. degrees, respectively, from Warsaw University of Technology in 1999 and from UPC in 2008. He is currently involved in the FP7 Strep Project DICONET, in the FP7 Network of Excellence BONE, and in the COST 2100 Action. His current research interests include optical networking with emphasis on network modeling, design, and performance analysis.

Davide Careglio received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2000 and 2005, respectively, and the Dr.Ing. degree in electrical engineering from Politecnico di Torino, Turin, Italy, in 2001. He is an Associate Professor in the Department of Computer Architecture at UPC. He is a member of the Advanced Broadband Communications Centre (CCABA) and of the Broadband Communications Research (CBA) group. He has recently been involved in several European Projects; he is currently involved in the FP7 Strep Project DICONET, and in the FP7 Network of Excellence BONE. His research interests are in the fields of all-optical networks with emphasis on packet based switching technologies, MAC protocols, quality of service (QoS) provisioning, and traffic engineering. He has participated in the technical program committees of several conferences, including IEEE ICC and IEEE Globecom.

Christian Gaumier received his Ph.D. from Ecole Polytechnique Fédérale de Lausanne (EPFL) in 1995. His doctoral research focused on modelling the propagation of signals over single-mode fibres in both linear and nonlinear regimes. Since 2001 he is director of the Telecommunications Laboratory of EPFL. His active research area includes dispersion compensation techniques, measurement techniques for fibre optics and dimensioning and performance analysis of core photonic communication networks. He participated to several European joint projects and is author and co-author of more than 50 publications in conferences journals and books. Dr. Gaumier is member of Communication Society of IEEE.

Josep Solé-Pareta obtained his M.Sc. degree in Telecom Engineering in 1984, and his Ph.D. in Computer Science in 1991, both from the Technical University of Catalonia (UPC). In 1984 he joined the Computer Architecture Department of UPC. Currently he is Full Professor with this department. He did a Postdoc stage (summers of 1993 and 1994) at the Georgia Institute of Technology. He is co-founder of the UPC-CCABA (http://www.ccaba.upc.edu/). His publications include several book chapters and more than 100 papers in relevant research journals (> 20), and refereed international conferences. His current research interests are in Nanonetworking Communications, Traffic Monitoring and Analysis and High Speed and Optical Networking, with emphasis on traffic engineering, traffic characterization, MAC protocols and QoS provisioning. He has participated in many European projects dealing with Computer Networking topics.