

QoS Multi-Spanning Tree for Optical Burst Switching

Joan Triay¹ and Cristina Cervelló-Pastor²

Department of Telematic Engineering, Technical University of Catalonia

Av. Canal Olímpic, 15, 08860, Castelldefels, Spain

¹joan.triay@upc.edu, ²cristina@entel.upc.edu

Abstract—Optical Burst Switching (OBS) has been proposed as a future high-speed switching technology for Internet Protocol (IP) networks that may be able to efficiently utilize extremely high capacity links without the need for optical-electronic (O/E) conversions at intermediate nodes. Among its benefits of supporting statistical multiplexing at a rational granularity, OBS still gives poor performance in terms of burst drop probability. In an Optical Burst Switched network, the existence of highly congested links could lead to unacceptable performance for the entire network, causing Quality of Service (QoS) degradation to flows inserted into the network. In this paper we propose a Multi-Spanning Tree architecture that improves QoS management by increasing the efficiency in the wavelength assignment process. The new protocol uses wavelength spanning trees on demand depending on the QoS constraints demanded by inserted traffics. Moreover, the protocol avoids burst overlapping on each edge of the spanning trees, while at the same time, permitting the reutilization of lambda assignments between nodes contained in the same branch. Therefore, the protocol allows traffic aggregation from many nodes into a single branch, improving efficiency and link utilization. We take a traffic engineering approach to branch selection with the objective of balancing the traffic across the network links to reduce congestion and to improve overall performance. In this paper, the architecture, the design of the protocol and some simulation results are presented.

I. INTRODUCTION

The Internet and thus the Internet Protocol (IP) are currently the basis for most of the information exchange all over the world. New services require high bandwidth, and new applications will appear as soon as higher bandwidth will be provided. In order to accommodate these capacity requirements, it seems that the intelligent optical network will be the only solution that can provide not only large capacity links in a flexible, dynamic and cost-effective way, but also to overcome any bottlenecks and limitations arising from the electronic processing. Optical Burst Switching (OBS) and Optical Packet Switching (OPS) are two such promising methods for transporting IP traffic directly over an optical wavelength division multiplexing (WDM) network [1].

Optical Burst Switching is a network technology that uses optics to switch data in a much smaller granularity resulting in a more efficient utilization of bandwidth [2]. In an optical burst-switched network, a data burst consisting of multiple packets is switched through the network all-optically. A control packet is sent in advance of the associated burst in order to reserve bandwidth and configure intermediate switches; so the burst can be forwarded transparently in the

optical domain. Hence, the OBS paradigm supports dynamic bandwidth allocation and statistical multiplexing of data.

Since optical burst-switched networks provide connectionless transport, there exists the possibility that bursts may contend with one another at intermediate nodes. Effective contention resolution is critical in OBS networks to restrict losses to a reasonably low level [3]. Wavelength conversion [4], Fiber Delay Lines (FDL) [5], deflection routing [6] and burst segmentation [7] are some of the techniques proposed to address the contention problem. Moreover, contention resolution schemes may be used to provide Quality of Service (QoS) in an all-optical core network [8].

In this paper we propose a protocol to provide an end-to-end QoS-IP service within an OBS network. We design a Multi-Spanning Tree protocol that improves aggregate throughput by increasing the efficiency in the wavelength assignment process. The new protocol combines a collision-free routing and wavelength assignment scheme with QoS constraints.

The remainder of the paper is organized as follows. Section 2 introduces the detailed design of the new protocol proposed in this paper. Section 3 presents the simulation results. Finally, Section 4 summarizes the main conclusions of the paper.

II. MULTI-SPANNING TREE WAVELENGTH ASSIGNMENT

The protocol presented in this paper creates multi-spanning trees of lambdas that later can be used in the wavelength assignment process in an Optical Burst Switched network. For this reason, the architecture has been named Multi-Spanning Tree Wavelength Assignment (MSTWA). The goal is to obtain a lower burst loss OBS network in comparison to use other wavelength selection schemes, such as the static ones [9]. We consider wavelength continuity constraint in the network. In addition, resource access restrictions can be used to offer QoS in a multiservice IP optical network.

First of all, we need to define some concepts. Fig. 1 may help to clarify the meaning of these terms. Trees in the protocol are associated with a root node and are composed of various branches that reach different destinations over the same wavelength (e.g. the black dotted links from root node number 1 of Fig. 1). Therefore, a branch defines a concrete reserved wavelength path between nodes contained in it (e.g. the *Branch1-11* of the figure). A source node can have multiple established trees to reach all necessary destinations (e.g. the black and the grey dotted line trees of the same figure).

In MSTWA, source nodes –the edge nodes of an OBS network– process and create tree branches of a specific wavelength depending on input traffics and their QoS constraints. Each edge node can generate multiple branches for each destination of its traffic flows. The establishment of a branch can collide with other branches rooted at other edge nodes and established over the same wavelength. In this case, the protocol offers recovery schemes to re-initiate the setup of the branch with another lambda. At the end, a given source will be root of multiple spanning trees over multiple wavelengths.

To offer the possibility of wavelength reusing between nodes in the same branch, inner nodes can also add their traffics to branches rooted at previous nodes in the tree. The addition of these traffics must be done under specific demands, so that the inserted bursts do not collide with bursts transmitted by previous nodes in the branch. Therefore, bursts transmitted by previous nodes are prioritized over the local burst transmissions when branches are being reused. The reuse of established trees will provide transmission chances for low-demand traffics or low burst loss probabilities for priority classes of service. Moreover, reutilization of established trees gives statistical multiplexing capabilities to our proposal.

A. Setup and burst transmission queries treatment

Algorithm on Fig. 2 resumes the procedure of processing a burst transmission in an edge node, which implies wavelength assignment. First, an initialization process executes several tasks such as updating the routing tables between endpoints, setting up the monitoring tables per node and initializing the list of available wavelengths in the network. In the current implementation, routing tables are filled with routes that conform the minimum spanning trees and with one or more alternative routes. Moreover, each source needs to know all the hops that conform the path between itself and the destination as the protocol uses source routing to signal the branch. It is important not to confuse between the routing spanning trees created at this phase and the multi-spanning wavelength trees that the protocol creates. Finally, the monitoring tables are

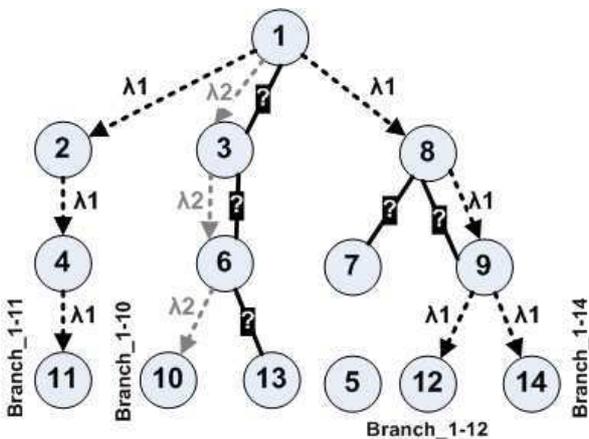


Fig. 1. Different branches over different trees rooted at node 1.

used to control the creation of new trees once an established one is overloaded.

Once the protocol is initialized in each node, burst transmission queries can be further processed. Previous to execute the reservation scheme at the source, the node checks the wavelength assignment availability to destination from its lambda routing tables and collects the burst's QoS constraints. The latter define to which QoS level flows can be assigned. Then a sorting algorithm is executed on the previous list of lambdas using the calculated QoS constraints to provide a prioritized order. The sorting principle will be explained later.

If the reservation has been successful, the burst is just scheduled to be transmitted at a concrete offset time and the control packet is immediately delivered to the next node. On the other hand, if the reservation has not been successful or the list of available lambdas becomes empty, the burst is rescheduled to be processed later if a maximum retry value is not reached. In this case, the burst is dropped.

In case the node has not any available wavelength to destination depending on the demanded QoS, it must initiate the MSTWA protocol to create a new lambda branch for future burst transmission queries. In the current implementation, among the three CoS defined in the network, the two highest priority ones can initiate the establishment of trees/branches.

B. Creation of multi-spanning wavelength trees

As previously pointed out, in case there is not any available wavelength to reach destination for a given QoS level, the source node should initiate a new wavelength branch. To signal and establish such branch, the node uses the MSTWA protocol.

The protocol makes use of a set of messages to signal and announce the establishment of multi-spanning wavelength

- 1: Initialization
- 2: New burst transmission query
- 3: $L_1 \langle \lambda \rangle = \text{getLambdaRoutes}(S, D)$
- 4: $\text{QoSReq}_1 = \text{getQoSRequirements}()$
- 5: $L_2 \langle \lambda \rangle = \text{sortingAlgorithm}(L_1 \langle \lambda \rangle, \text{QoSReq}_1)$
- 6: **if** $L_2 \langle \lambda \rangle$ is not empty **then**
- 7: **repeat**
- 8: $\lambda_i = \text{nextWavelength}(L_1 \langle \lambda \rangle)$
- 9: $\text{reserve} = \text{checkReservation}(\lambda_i)$
- 10: **until** reserve is true AND $L_2 \langle \lambda \rangle$ is not empty
- 11: **if** reserve is false **then**
- 12: $\text{try} \leftarrow \text{try} + 1$
- 13: **if** try is less than maxtry **then**
- 14: Reschedule burst transmission query
- 15: **else**
- 16: Drop transmission query
- 17: **end if**
- 18: **end if**
- 19: **else if** QoSReq_1 is high priority **then**
- 20: Initiate branch setup with MSTWA
- 21: **end if**

Fig. 2. General algorithm to process burst transmission queries.

trees. These messages are: the Optical Burst Setup Data Unit (OBSDU), the Optical Burst Release Data Unit (OBRDU), the Acknowledge (ACK) and the Negative Acknowledge (NACK). Besides previous messages, Control Packets or Burst Header Packets (BHP) carry information about the QoS level assigned to the burst. This extra data, apart from current offset and burst length data, specifies how bursts are treated inside the network in terms of QoS.

The MSTWA protocol is initiated by a new branch reserve query composed of various parameters. From the source and destination values, the node gets the list of routes, included the shortest one that conforms the minimum spanning tree. It also queries to the routed wavelengths table the list of branches that the node can use to execute the new distributed branch setup. The choice of wavelengths will be explained later.

Suppose, for example, that the *node 1* in Fig. 1 needs to setup a lambda branch to destination *node 13* (e.g. the branch with λ_1). From the list of wavelengths, the node gets the first entry, creates a new OBSDU message and transmits it to the next node along the path (e.g. *node 3*). The OBSDU is then processed by the following nodes. Each of these nodes has to check if the new queried branch collides with a previously established one. What is mainly checked in this step is the loop free requirement for a given wavelength. The loop free per wavelength will prevent in core nodes burst collisions from different previous hop nodes (i.e. input interfaces), in case bursts use the same wavelength. Loops are only permitted when the current node is in fact the destination node. In this case, it is assumed that there is no burst blocking. If the current branch collides with a previous one, the node that is currently processing the message must generate a NACK message. This message notifies the source node and root of the branch that the establishment of the branch has not been successful, and specifies the reason of the collision. Fig. 1 shows an example of possible collision on the establishment of the branch. *Node 8* must reject the request of branch establishment if the wavelength queried by *node 7* to reach a node farther from *node 8* is already used by *node 1* from a different input link, e.g. λ_1 in the figure.

On the other hand, if there is no collision in the wavelength branch establishment, the node prepares an entry in its database and marks it with the state *Processing*. Then, immediately, the current node transmits the OBSDU message to the next node along the path to destination.

If the OBSDU message finally arrives to destination, and the wavelength loop free is still satisfied, the destination node marks the branch entry in its database as *Established*. Then it generates a new ACK message to positively acknowledge to source that the branch has been successfully established. The ACK message is processed by the nodes on the reverse path, from destination to source, changing the previously established entry in their database from *Processing* to *Established*. Core nodes of the branch also update their lambda routing tables with the new branch.

At the end, the ACK message arrives to the source/root node of the branch. At this moment, the branch changes its state to

Established and the MSTWA protocol finishes. From now to forward the source node has the use of a new wavelength to reach a given destination.

C. Wavelength selection on Tree/Branch establishment

When a source node needs to create a new tree or branch it follows a wavelength selection criterion in order to improve the probability of success. If the node does not have any tree rooted at itself, then the wavelength choice is randomly done from the list of available wavelengths on the network. Otherwise, the node tries to establish the new branch using a known lambda.

In the choice criteria, the root node sorts the list of lambdas on which to initiate the query. Firstly, it fills the list with the set of lambdas classified as *longerTreeType*. These wavelengths are got from the established trees for which the current branch path shares –in order– the maximum number of links. For example, suppose there is a branch for the wavelength λ_0 following the path between *A* and *D*, $A \rightarrow B \rightarrow C \rightarrow D$, and a new query to destination *E* has to be processed. If the path between *A* and *E* is $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, a good wavelength candidate would be the same used for the established branch. Between nodes *A* and *D* the branch establishment will be successful as it does not collide with the previous ones, and it will only last the link between *D* and *E* to check. The probabilities of success will be higher in comparison to the use of any other random lambda for which the previous hop collision checking process must be positively passed at each hop.

The second class of wavelengths are the *branchTreeType* ones. This type of lambdas try to extend the use of the same lambda at root nodes. In this case, the root node tries to construct the minimum number of trees to reach all the necessary destinations. This strategy will leave more lambdas to other source nodes of the network to establish their own trees, obtaining a fairer distribution of wavelengths in the network and a better utilization of the network's resources. Another example follows: suppose the same established branch of the previous example (λ_0 over $A \rightarrow B \rightarrow C \rightarrow D$). The new branch to process is $A \rightarrow B \rightarrow C \rightarrow E$. As it can be seen, the second path is a new branch rooted by the same node and nodes *B* and *C* are common to both branches. Therefore, the setup of the new branch over λ_0 will be the better choice.

D. Wavelength selection principle on burst transmission

In steady state, nodes may have some trees rooted by themselves. In a non-multiservice scenario, the reservations over the lambdas of these trees shall have more probabilities of success if all the network nodes use the same principle of “trying first wavelengths of their own trees”. But it can happen that these lambdas are also under high load and the reservation is not possible. In order to solve this, the algorithm provides a way to reuse wavelengths of trees for which the current node is not the root node. Therefore, the established trees are not unique to their root nodes, allowing other nodes to overlap bursts over these wavelengths. This gives statistical multiplexing capabilities to our proposal. In order to make

this overlapping method efficient there is the need of delaying the bursts from previous hops using Fiber Delay Lines (FDLs). These are used when an intermediate node in a branch reserves resources to transmit bursts over a wavelength occupied at the same period of time by a downstream node.

E. Wavelength and class of service assignment with MSTWA on a multiservice scenario

Branch reutilization can be used to offer QoS in a multi-service scenario by restricting the access to lambda resources according to the demanded QoS. The classification and sorting of lambdas is done by an algorithm depending on different classes of service (see Table I).

In the current implementation of the protocol, three classes of service (CoS) have been taken into account. The highest priority class uses wavelengths provided by the MSTWA scheme. To achieve a lower burst loss probability, the nodes use for this CoS rooted and non-rooted branches in the wavelength assignment process. This gives a longer list of lambdas, and therefore, the reservation probability is expected to be higher.

The second highest priority CoS also uses our tree scheme, but in this case, flows can only be assigned to branches rooted by the node itself (the edge node that is inserting the burst traffic into the OBS network). If these are the only possible wavelengths, the node still has a chance to reserve resources for the burst transmission, but not so many possibilities as the highest priority CoS. Apart from this, bursts with this kind of CoS can only be interfered by *Class 1* bursts transmitted over non-rooted branches. This should provide differentiated behaviors between the two highest priority classes.

Finally, the lowest priority CoS does not use MSTWA. In this case, the wavelength is randomly chosen from the list of available lambdas (e.g. lambdas that have not been assigned to branches/trees by the node itself, or other nodes of the network) and the reservation in core nodes is checked according to this level of priority.

III. RESULTS

Simulations results check the correct operation of the protocol and compare it with a static random wavelength assignment protocol using the JET reservation scheme [10]. The current paper is focused on the analysis of the burst blocking probability and the calculation of the burst queue delay at source nodes depending on the QoS level demanded by ingress traffic flows.

The simulator used in the tests has been done from scratch using the DESMO-J framework [11].

TABLE I
CLASSES OF SERVICE AND THEIR MAPPING TO MSTWA.

CoS	Priority	Scheme	Branch selection
Class 1	Highest	MSTWA	Root + Non-root
Class 2	Medium	MSTWA	Root
Class 3 (BE)	Lowest	Other	Random

For the simulations, the NSFNET with 14 nodes has been used as an OBS network. The capacity per λ is of 1 Gbps. The number of wavelengths vary among simulations between 16, 32 and 64. The simulations presented here assume no wavelength conversion capability in the network.

In all the simulations, OBS edge nodes attend traffic queries of 470 Mbps from their sources. Client data packets are of 1500 bytes of length and edge nodes assemble bursts of 100,000 bytes. The arrival of burst transmission queries is supposed to follow a Poisson Distribution with the time needed to assemble the burst as the mean parameter.

Simulation results have been calculated from running the simulations 20 times. The average probability value is a mean with a 95% confidence interval. Not all the graphs include these intervals in order to improve the readability.

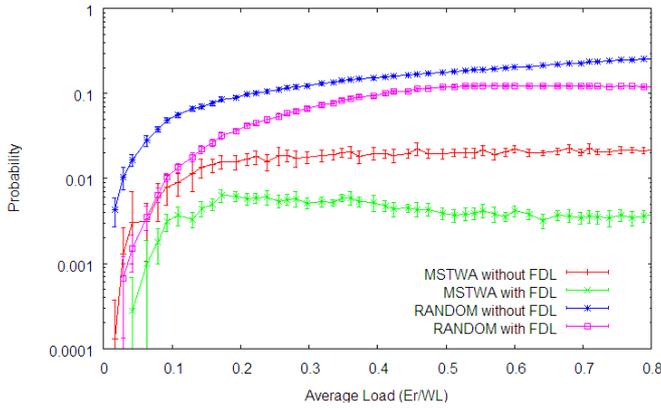
A. Burst blocking probability

These results concern to the burst blocking probability under different network load scenarios. Two algorithms, each one with two variants (use or absence of FDLs) are compared, JET with random wavelength selection (which we call it simply RANDOM in the graphs) and JET integrated with our protocol proposal (named MSTWA in the graphs).

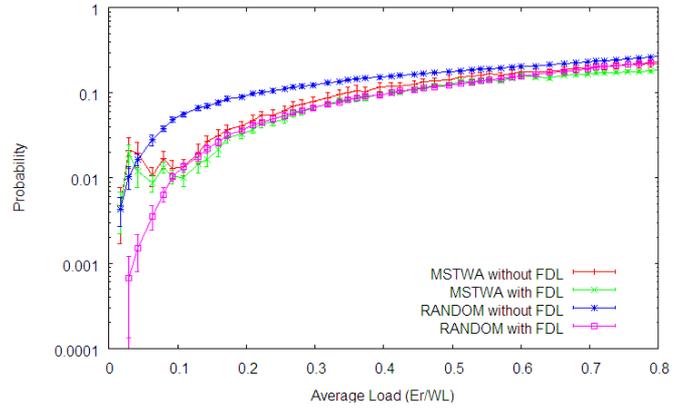
Fig. 3 groups graphs about the protocol performance in absence of QoS definitions. As it can be seen in subfigure 3(a), MSTWA performs better than the use of the other assignment protocol on almost all the load range. This behavior can be explained by the nearly non-blocking nature of the multi-tree wavelength approach. Moreover, at low loads, as the branches are not highly overloaded, core nodes can efficiently transmit their bursts over branches rooted at other nodes.

For the random protocol, burst losses occur mainly in switching on core nodes, and some in transmission. Alternatively, in MSTWA, losses at high loads occur on transmission (i.e. at the source edge node). These blockings have not been taken into account on the previous subfigure, and are shown in 3(b), which provides the total burst blocking probability. Under heavy loads, if current wavelength trees are overloaded, burst transmissions will not be able of reserving enough resources. The node will then try to initiate the creation of another branch, but because of the high load that occurs in most of the nodes, the list of available wavelengths is scarce. This causes that the branch creation is unsuccessful. Previous argument explains the increase of transmission burst blocking probability in our protocol as shown in subfigure 3(b), although its value is better than the value given by the RANDOM scheme, specially at medium-high load.

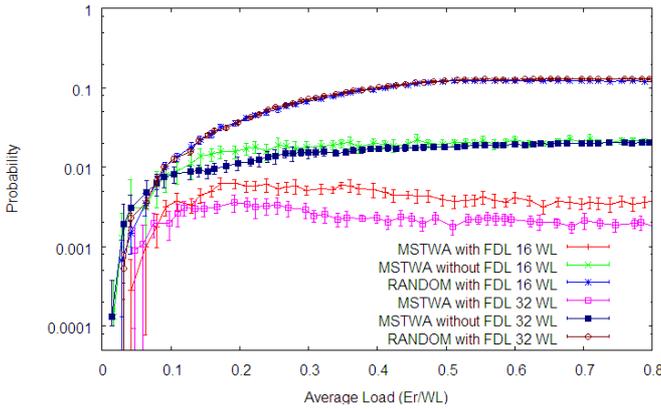
Fig. 3(c) shows the burst blocking probability relationship between scenarios with different number of wavelengths (16, 32). MSTWA improves the efficiency in comparison to the RANDOM scheme for the two scenarios. Moreover, evaluating the performance of the protocol when the number of wavelengths increases, it can be seen a slight increment of performance, specially when the protocol uses FDLs. With 32 lambdas and FDLs, the network achieves a fairer tree distribution offering better results.



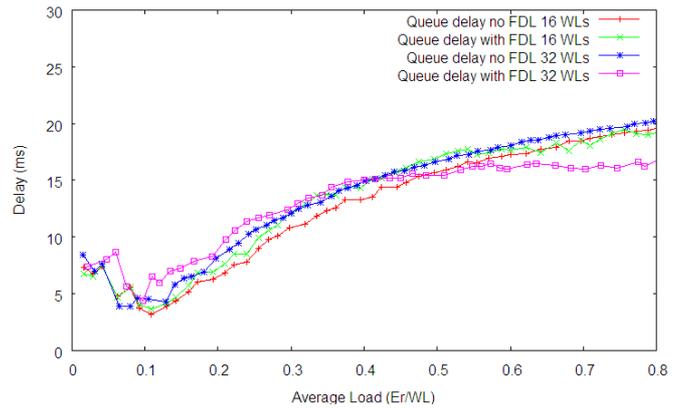
(a) Burst blocking probability in network core with 16 wavelengths.



(b) Total burst blocking probability with 16 wavelengths.



(c) Burst blocking probability in network core for different number of wavelengths.



(d) Burst queue delay for 16 and 32 wavelengths using MSTWA.

Fig. 3. Results without QoS.

B. Burst query queue delay

One of the requirements of the protocol to transmit a burst is to have the use of a routed wavelength via an MSTWA tree. If the current node does not have any wavelength to reach the destination, then it must initiate the MSTWA tree establishment. Between the arrival of the burst transmission query and the availability of a lightpath to transmit the burst, there is some delay to create the tree. This delay causes the burst to remain buffered at the source node. Moreover, this delay can incur in losses of bursts if buffers are filled too fast, or in errors if the delay does not satisfy the QoS of the application or higher transport layer protocols (e.g. TCP or RTSP) at source nodes.

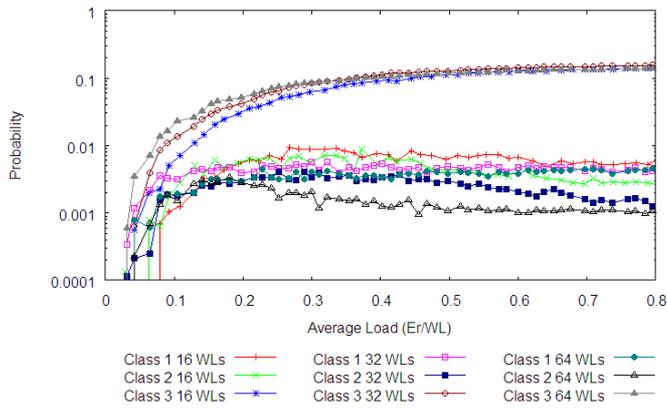
Fig. 3(d) shows the delay on the burst queues against network load when the network is using 16 or 32 wavelengths. The graph values for high loads show how the saturation and scarce of new wavelength trees to attend the new demands of traffic increases the queue delay. Some traffics will do not have wavelengths over which to transmit the bursts, and therefore, these bursts remain in queue until they are dropped—because the maximum transmission retry value is reached—. It can also be seen that at low loads, queue delay is slightly greater than at low-medium values. The establishment of the

first trees imply a first big delay that slightly decreases when the subsequently transmission queries can be satisfied without the need of creation of new trees. From this point forward, delay increases.

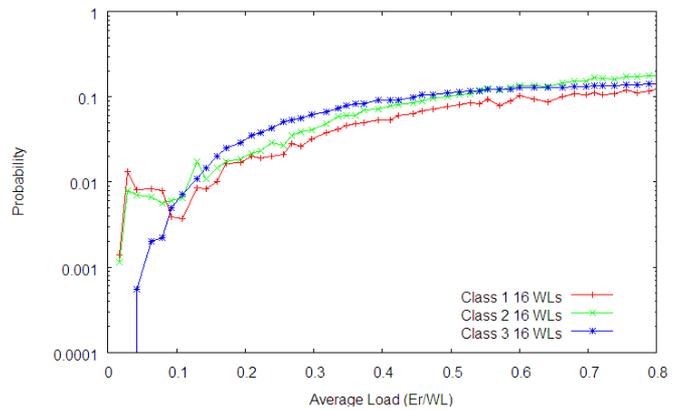
C. MSTWA with QoS results

Finally, some results about the application of MSTWA to provide QoS in a multiservice environment are presented in this section. These simulations were modified in respect of the previous ones to include traffic flows of different CoS.

Fig. 4(a) shows comparative results of burst blocking probability in the network core for each CoS in three different scenarios—16, 32 and 64 wavelengths—. For all the load range, higher priority classes experience a lower burst blocking probability than *Class 3*. *Class 1* and *Class 2* get similar results, though the second one experiences a better performance in each of the three scenarios. These results should not be strange as both classes use first the set of their own rooted wavelengths and they have the capability to initiate new wavelength branches if there are not enough resources at that specific moment. The different performance between the first two classes comes from the fact that a *Class 2* burst can only collide with *Class 1* bursts transmitted over non-rooted branches, while *Class 1* can collide with same class



(a) Comparative burst blocking probability per CoS for different number of wavelengths.



(b) Total burst blocking probability per CoS using 16 wavelengths.

Fig. 4. Results with QoS.

bursts transmitted either over rooted and non-rooted branches. Nevertheless, the probability remains under the 0.01 for any number of wavelengths. This special case could be used to provide fixed QoS to certain flows for different network scenarios. For higher priority classes, the more the number of wavelengths, the more the increase in performance. This follows the same behavior given on Fig. 3(c).

Finally, Fig. 4(b) gives a comparison between classes with a network scenario of 16 wavelengths in terms of total burst blocking probability. In general, *Class 1* obtains better results than *Class 2* due to the fact that the first class has a longer list of available wavelength branches that can be used in the wavelength assignment process.

IV. CONCLUSION

This paper has described a new Multi-Spanning Tree architecture (MSTWA) that improves aggregate throughput in Optical Burst Switched networks by increasing the efficiency in the wavelength assignment process. The protocol can be used to provide QoS in a multiservice environment by restricting the access to lambda resources. The creation of the wavelength spanning trees is done on demand, and these avoid burst collisions in core networks by establishing a collision free wavelength distribution. Statistical multiplexing is offered by means of overlapping bursts from nodes contained in the same branch.

On one hand, simulation results show an increase on efficiency in the burst blocking probability with regard to the use of a common resource reservation protocol such as JET with random wavelength assignment. On the other hand, burst queueing delay at edge nodes increases since in our case, transmission of bursts require a successful established wavelength tree with enough capacity. In terms of QoS, simulations results show that with MSTWA, higher priority traffic flows can get constraint-guaranteed services for any number of wavelengths.

Although the performance of all simulations give positive results, new methods to decrease burst collisions and burst droppings due to unavailable wavelength trees on the edge ingress node –when the number of lambdas is scarce– must

be implemented in future research tasks. Moreover, new class of service mappings should be studied to provide a broader range of possible CoS.

ACKNOWLEDGMENT

Work supported by the Spanish “Ministerio de Educación y Ciencia” and FEDER under project TSI2006-12507-C03-03, the i2CAT Foundation and by the Department of Education and University of the Government of Catalonia and the European Social Funding with a pre-doctoral grant.

REFERENCES

- [1] S. Yao, S. J. B. Yoo, B. Mukherjee, and S. Dixit, “All-optical packet switching for metropolitan area networks: Opportunities and challenges,” *IEEE Commun. Mag.*, vol. 39, no. 3, pp. 142–148, Mar. 2001.
- [2] Y. Chen, C. Qiao, and X. Yu, “Optical Burst Switching: a new area in optical networking research,” *IEEE Network*, vol. 18, no. 3, pp. 16–23, May 2004.
- [3] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, “A unified study of contention resolution schemes in optical packet-switched networks,” *J. Lightwave Technol.*, vol. 21, no. 3, pp. 672–683, Mar. 2003.
- [4] Z. Rosberg, A. Zalesky, H. Vu, and M. Zukerman, “Analysis of OBS Networks With Limited Wavelength Conversion,” *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 1118–1127, Oct. 2006.
- [5] I. Chlamtac, A. Fumagalli, and C. J. Suh, “Multibuffer delay line architectures for efficient contention resolution in optical switching nodes,” *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2089–2098, Dec. 2000.
- [6] Y. Chen, H. Wu, D. Xu, and C. Qiao, “Performance analysis of optical burst switched nodes with deflection routing,” in *Proc. IEEE ICC’03*, May 2003, pp. 1355–1359.
- [7] H. C. Cankaya, S. Charcraonon, and T. S. El-Bawab, “A preemptive scheduling technique for OBS networks with service differentiation,” in *Proc. IEEE Globecom’03*, Dec. 2003, pp. 2689–2673.
- [8] M. Yoo, C. Qiao, and S. Dixit, “QoS performance of Optical Burst Switching in IP-over-WDM networks,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 2062–2071, Oct. 2000.
- [9] J. Teng and G. N. Rouskas, “Wavelength Selection in OBS Networks using Traffic Engineering and Priority-based Concepts,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 8, pp. 1658–1669, Aug. 2005.
- [10] M. Yoo and C. Qiao, “Just-Enough-Time (JET): a high speed protocol for bursty traffic in optical networks,” in *Proc. of IEEE/LEOS conf. on Tech. for a Global Information Infrastructure*, Aug. 1999, pp. 26–27.
- [11] University of Hamburg - Department of Computer Science. DESMO-J (Java): A Framework for Discrete-Event Modelling and Simulation. [Online]. Available: <http://www.desmoj.de>