Relaxing Delayed Reservations: An approach for Quality of Service differentiation in Optical Burst Switching networks

Kostas Christodoulopoulos, Kyriakos Vlachos, Kostas Yiannopoulos and Emmanuel A. Varvarigos

Computer Engineering and Informatics Dept. (CEID) and
Research Academic Computer Technology Institute (RACTI), University of Patras
GR26500, Rio, (e-mail: kvlachos@ceid.upatras.gr)

***Abstract*— In this paper we present a signaling protocol for QoS differentiation suitable for optical burst switching networks. The proposed protocol is a two-way reservation scheme that employs *delayed* and *in-advance* reservation of resources. In this scheme delayed reservations may be *relaxed*, introducing a *reservation duration* parameter that is negotiated during call setup phase. This feature allows bursts to reserve resources beyond their actual size to increase their successful forwarding probability and is used to provide QoS differentiation. The proposed signaling protocol offers a low blocking probability for bursts that can tolerate the round-trip delay required for the reservations. We present the main features of the protocol and describe in detail timing considerations regarding the call setup and the reservation process. We also describe several methods for choosing the protocol parameters so as to optimize performance and present corresponding evaluation results. Furthermore, we compare the performance of the proposed protocol against that of two other typical reservation protocols, a Tell-and-Wait and a Tell-and-Go protocol.**

***Index Terms*— Optical burst switching, signaling protocol, optical networks, Quality of Service.**

I. INTRODUCTION

Switching in core optical networks is currently performed using high-speed electronic or all-optical circuit switches. Switching with high-speed electronics requires optical-to-electronic (O/E) conversion of the data stream, making the switch a potential bottleneck of the network: any effort (including parallelization) for electronics to approach the optical speeds seems to be already reaching its practical limits. Furthermore, the store-and-forward approach of packet-switching does not seem suitable for all-optical implementation due to the lack of practical optical Random-Access-Memories to buffer and resolve contentions. Circuit switching on the other hand, involves a pre-transmission delay for call setup and requires the aggregation of microflows into circuits, sacrificing the granularity and the control over individual flows and their QoS requirements. Especially for bursty traffic, circuit switching is known to be inefficient.

Optical burst switching (OBS) [1] has been introduced to combine the advantages of both packet and circuit switching and is considered a promising technology for the next generation optical Internet. An OBS network consists of a set of *optical core* routers and *edge routers*. An optical burst is constructed at the network edge, by aggregating a number of variable size packets. In general, each edge router maintains a separate (virtual) queue for each Forwarding Equivalence Class (FEC) to hold the data packets that belong to that FEC until a burst is formed (A FEC is defined from a source-destination pair and optionally from a set of Quality-of-Service requirements).

A number of signaling protocols [2]-[10] and QoS schemes [12]-[17] for OBS networks have been proposed so far. The signaling schemes found in the literature can be categorized into two main classes: two- and one-way protocols. In two-way reservation schemes (also called *Tell-and-Wait*), end-to-end connections are fully established before the transmission of any data can start, while resources at intermediate nodes are reserved immediately upon the arrival of the SETUP packet at these nodes. Recent research efforts like the WR-OBS [2],[3], have shown that such reservation schemes can enable the implementation of a bufferless core network with limited node wavelength conversion capability by moving the processing and buffering operations at the edge.

In one-way reservation schemes (also called *Tell-and-Go*), a setup packet is sent in advance over the path, preceding the arrival of the burst by a small time offset. This minimizes the pre-transmission delay, but can result in high burst dropping probability. A number of one-way reservation schemes have been proposed for OBS networks, including the Ready-to-Go Virtual Circuit protocol [6], Just-Enough-Time (JET) [7], Horizon [8],[9] and Just-In-Time (JIT) [5],[10]. The differences among these variances lie mainly in the time instances that determine the allocation and the release of the resources. Furthermore, for the one-way schemes that employ delayed reservations, sophisticated channel scheduling and void filling algorithms have been proposed to resolve contentions and efficiently utilize the available bandwidth [9],

1-4244-0425-8/06/$20.00 ©2006 IEEE

[11]. Although very promising, one way schemes rely on wavelength conversion to resolve contentions, which poses specific hardware requirements, raising issues such as node scalability, size and cost.

In order to enable QoS provision and service differentiation in OBS networks various one-way schemes have been introduced, including the *JIT offset-time-based* scheme that uses time offsets to isolate different classes of traffic [13], the *composite-burst assembly* scheme that mixes traffic classes during burst assembly and provides QoS via prioritized burst segmentation [14], the *preemptive wavelength reservation* mechanism, where each class is associated with a predefined usage limit [15], the *early dropping* mechanism that randomly drops bursts depending on their class [16], and finally the *FRR* scheme, where burst length prediction is combined with the *JIT offset-time-based* scheme [17].

Having identified the major advantages and weaknesses of the two complementary classes of protocols, hybrid signaling schemes have been especially designed to combine features from both classes. [4] introduces a hybrid scheme that employs Tell-and-Wait reservations up to an intermediate node followed by an (unacknowledged) one-way reservation process until the egress node. This hybrid protocol provides a trade-off between burst loss and delay (by selecting the intermediate node) and thus enables QoS differentiation.

In this paper, we propose the *Efficient Burst Reservation Protocol* (abbreviated EBRP). EBRP is suitable for bufferless Optical Burst Switching networks and exploits the advantages of both classes of protocols to achieve efficient burst-level reservations and to provide QoS differentiation. EBRP is a two-way scheme, but, unlike typical Tell-and-Wait schemes, reserves resources only for a given duration (*timed*/*delayed reservation*) similarly to the one-way schemes. However, in the EBRP protocol delayed reservations may be "*relaxed*", through the introduction of a *reservation duration* parameter that may exceed the burst duration. EBRP uses this feature to increase the successful forwarding probability and also to provide service differentiation by assigning different reservation duration parameters to different classes of traffic. The performance results we obtained show that the proposed scheme gives a low dropping probability for bursts that can tolerate the round trip delay required by the two-way reservation process.

The remainder of the paper is organized as follows. Section II describes the main features of the protocol, while Section III describes in detail the burst reservation process and the corresponding timing considerations. Section IV presents various methods of choosing the parameters of the protocol. Section V presents the parameters optimization results, assesses the protocol performance by comparing it to that of typical Tell-and-Wait and typical Tell-and-Go schemes, and investigates its performance when multiple classes of service are employed.

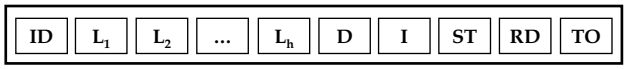

Figure 1: The different fields of the SETUP packet

II. Protocol Main Features

The challenges in optical burst switching networks that employ no or minimum buffering in the core, is how to achieve efficient usage of resources (that is, to consume the reserved resources only when they are actually needed) and how to provide class differentiation, with limited control plane overhead. In order to achieve these objectives, the EBRP protocol employs two basic features to schedule the bursts: *Relaxed Delayed Reservations* and *In-Advance Reservations*.

With *Relaxed Delayed Reservations*, outgoing capacity is reserved for some specific duration and released after the data has passed through the node. Unlike [7], however, in EBRP the requested reservation period may exceed the actual burst transmission time during the call setup phase, in order to increase the acceptance probability at subsequent nodes. In this case the strict time requirements are restored during the acknowledge phase. The extra reservation duration at an intermediate node facilitates the scheduling of the burst in the time domain at downstream nodes. By controlling the degree of reservation flexibility, EBRP can provide QoS differentiation.

With *In-Advance Reservation*s EBRP can reserve capacity in the future, at the first time it becomes available, if it is not available when requested. If sufficient available capacity cannot be found within the maximum delay requirements of the burst, the request is rejected. By allowing future reservations, bursts have higher probability of being scheduled in a setup phase, avoiding the control overhead and delay associated with call setup repetition.

In order to employ these mechanisms, the burst size has to be known and communicated during the connection setup process. Moreover, in order to schedule or reject new requests, each node has to be aware of its own resource availability by maintaining a *capacity utilization profile* of its outgoing links.

The proposed two-way scheme employs two control packets for call setup, namely: a SETUP and an ACK/REJECT packet. The SETUP packet is transmitted from source to destination and is used for resource negotiation. If the reservation is successful, an ACK message is sent back to the source to confirm the delayed reservation. The ACK packet is a replica of the SETUP packet, when the latter reaches its destination node, and communicates to intermediate nodes the (agreed) time intervals for which the resources have been allocated. On the contrary, if the reservation process is blocked at an intermediate node, a REJECT packet is generated and sent backwards to release the capacity reserved at intermediate nodes and notify the source. The use of delayed reservations relieves the network from the additional control plane overhead associated with the tearing

down of the reservations.

Figure 1 shows the fields of the SETUP packet in the EBRP protocol. The path of the SETUP packet can be specified as a sequence of link identifiers $L_1,L_2,...,L_h$, corresponding to the links that this packet must traverse (*source routing*). Each node reads the first link identifier to determine the outgoing link to which it should be routed, and cyclically rotates the link identifiers so that the one just read becomes last. Basic fields of the SETUP packet that must be communicated to all nodes is the relaxed reservation duration *RD*, the information size *I*, the requested starting time *ST*, the time-offset *TO* and the delay tolerance *D*:

- The start time field *ST* specifies the time at which the reservation of capacity for the specific outgoing link should begin. *ST* is relative to the arrival time of the SETUP packet at the node. Therefore, the *ST* field is initially set equal to the round trip delay time (T_{RTT}) and is updated at the intermediate nodes according to their resource availability, as described in the following section.
- The time-offset field *TO* contains the time, following the reception of the ACK packet at the source, after which the source should start transmitting the burst. The *TO* field is updated at every node in a way to be described later.
- The information size field *I* specifies the amount of information (in bits) that will be transmitted.
- The delay tolerance *D* specifies the maximum allowable delay for the burst. Clearly, we must have $D > T_{RTT}$, otherwise the requested transmission cannot be served within the desired deadline over that path.
- The reservation duration time field *RD* specifies the maximum time period following *ST*, during which the specific outgoing link should be reserved. The *RD* field has to be set at least equal to the burst transmission time ($T_{data}= I / C$, where *C* is the link capacity), but it can be larger than that. The *RD* field provides control over the allowed degree of the delayed reservation mechanism. For example, if the *RD* field is initially set equal to the burst transmission time T_{data} , then resources are reserved exactly for the time needed, while when *RD* exceeds the burst transmission time, a more "*relaxed*" delayed reservation is made, providing more flexibility for the reservations that have to be made at downstream nodes. If a SETUP packet reserves bandwidth at a link for a duration larger than the burst transmission time, it is given a higher probability of reserving at least the minimum required duration at subsequent, downstream nodes. The actual reservation period is restored to the burst transmission time T_{data} during the acknowledgement phase. The way the *RD* field is updated is presented in the following section.

III. Burst Reservation Process

Figure 2 illustrates the timing considerations of the EBRP protocol, where a set up process is instantiated between nodes S_0, S_1, S_2 and S_h. In particular, Figure 2 (a) illustrates the case when a request is blocked at an intermediate node, while Figures 2 (b) and (c) show the call setup and acknowledgment

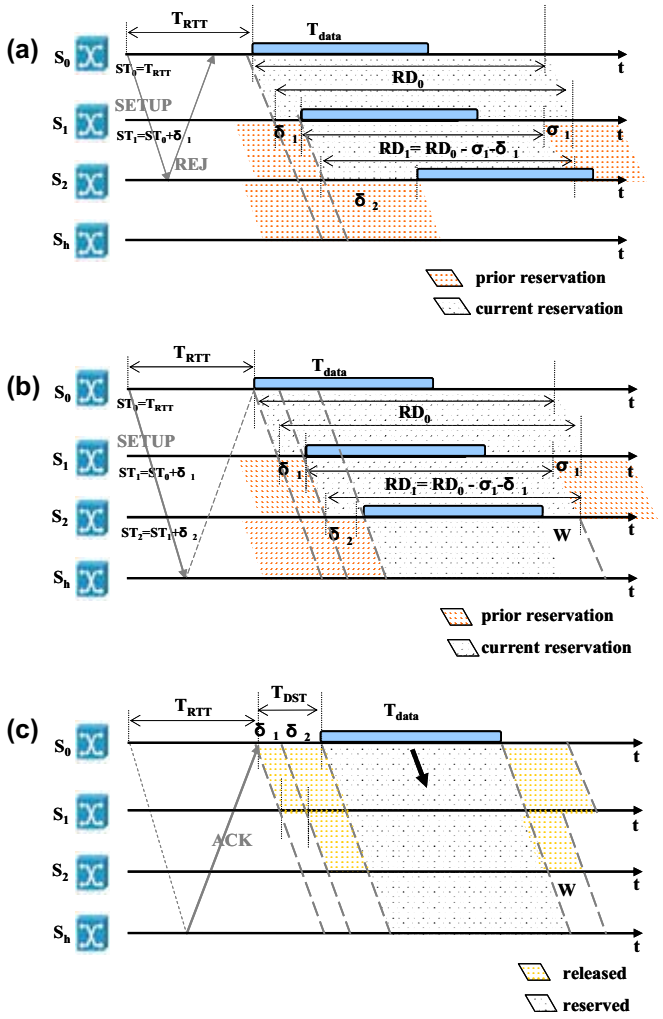

Figure 2: Timing considerations in the proposed signaling scheme. (a) Blocked call setup phase at an intermediate node, (b) successful end-to-end call setup and (c) acknowledgement phase, where the excess resources are released.

phases of a successful reservation.

Let ST_{i-1}, TO_{i-1}, RD_{i-1} be the values of the fields *ST*, *TO*, *RD* when the SETUP packet reaches node S_i (i=1,2,...,h-1). When node S_i receives the SETUP packet, it finds the first time t^i_{start} relative to the SETUP packet arrival that $t^i_{start} \geq ST_{i-1}$, and enough residual capacity is available to accommodate the burst. In order to do so, capacity should be available for the time period $\left[t^i_{start}, t^i_{start} + T_{data}\right]$ and $t^i_{start} + T_{data} \leq ST_{i-1} + RD_{i-1}$.

- We denote $\delta_i = t^i_{start} - ST_{i-1}$, where $\delta_i \geq 0$. If a t^i_{start} that satisfies the above requirements can be found, node S_i reserves the resources starting from time $t^i_{start} = ST_{i-1} + \delta_i$ up to time $t^i_{end} = \min\left[(ST_{i-1} + RD_{i-1}), t^i_{available}\right]$. We denote by $t^i_{available}$ the time that capacity stops to be available due to reservations made by other bursts.
- In the case that $t^i_{start} + T_{data} > ST_{i-1} + RD_{i-1}$ the request is blocked and a REJECT packet is sent over the reverse path to release the reserved resources and notify the source.

In the case of a successful reservation, node S_i updates the fields of the SETUP packet and forwards it to the next node. In particular, it updates the reservation starting time, time offset and reservation duration fields carried by the SETUP packet as follows:

$$ST_i = ST_{i-1} + \delta_i, \quad (1)$$

$$TO_i = TO_{i-1} + \delta_i, \quad (2)$$

$$RD_i = RD_{i-1} - (\delta_i + \sigma_i). \quad (3)$$

where $\sigma_i = \max\left(0, \left[(ST_{i-1} + RD_{i-1}) - t^i_{available}\right]\right)$. In other words σ_i corresponds to the decrease of the reservation duration field at the trailing edge due to other reservation, while in total the RD_i field is decremented by $\sigma_i + \delta_i$.

In Figure 2 (a), the first node S_1 on the path finds that the earliest time at which enough bandwidth is available is an offset time δ_1 later than the time it was requested, it updates the [*ST*, *TO*, *RD*] fields, from [T_{RTT}, 0, RD_0] to [$T_{RTT}+\delta_1$, δ_1, $RD_0-(\delta_1+\sigma_1)$] and forwards the SETUP to node S_2. Similarly, node S_2 finds that the earliest time at which resources are available is after an offset time δ_2. However, reservation (void filling in this case) cannot be performed since no period of duration T_{data} exists and S_2 drops the request. In the example depicted in Figure 2 (b), S_2 finds adequate capacity beyond the δ_2 time offset and grants the request. In the acknowledgement phase (Figure 2 (c)) the excess resources reserved at nodes S_1 and S_0 are released.

When the SETUP packet arrives at the last core node prior to the egress router, there is no need to reserve resources beyond the burst duration (assuming that the egress router is commissioned only to buffer and disassemble the bursts). Thus, node S_{h-1} (node S_2 in Figure 2) reserves resources only for duration equal to the burst transmission time that is for the period: $\left[t^{h-1}_{start}, t^{h-1}_{start} + T_{data}\right]$.

The SETUP packet that reaches the egress router has accumulated all the time offsets δ_i issued by the intermediate nodes and therefore ST_{h-1} determines the earliest transmission starting time for which resources are available. The destination node S_h (possibly after checking for the availability of adequate memory to store the specified burst size), acknowledges the successful reservations by sending an ACK packet back to the source. The ACK packet contains the fields:

$$TO_{h-1} = \sum_{i=0}^{h-1} \delta_i, \quad (4)$$

$$ST_{h-1} = T_{RTT} + \sum_{i=0}^{h-1} \delta_i \quad (5)$$

Upon receiving the ACK packet, the intermediate nodes retrieve the agreed transmission starting time ST_{h-1}, update their reservations to exactly match the transmission of the burst duration time T_{data} (enforcing/restoring strict delayed requirements), and release the remaining of the resources. Upon receiving the ACK packet, the source waits for time equal to *TO* and begins transmission. Note that by the way the reservations were made, if the source starts transmitting the burst at a time *TO* after the reception of the ACK packet, the burst is guaranteed to find available capacity for duration equal to T_{data} at all intermediate links of the path when it arrives at these links.

If buffering at the destination node of the core network is also a limited resource, a (timed/delayed) buffer reservation also has to be made. The buffer required at the destination node can be viewed as the last leg of the reservation and can be treated in the same way bandwidth is treated.

IV. Reservation Duration Functions

The reservation duration (*RD*) field carried in the SETUP message depends on the resource availability windows found by the SETUP message at previous nodes. During the downstream propagation of the SETUP packet, the *RD* field is trimmed down and is finally set equally to the burst transmission time. If an intermediate node cannot grant capacity for a period at least equal to the burst transmission time T_{data} inside the specified *RD* period, the setup process is rejected. Large values of the *RD* increase the flexibility in the reservation process and provide higher probability of reserving at least the minimum required duration at subsequent nodes. For traffic differentiation, each Class of Service (CoS) can be mapped to a different *RD* value, or more general to a different *RD* function. Initializing the *RD* field with a large value for a certain FEC increases the priority of that FEC at the expense, however, of a degradation in the performance experienced by other FECs.

The initial value of the *RD* field should not be a static per FEC parameter, but has to depend on the burst size and the number of hops, since these parameters affect the trimming process of the *RD* window during the reservation process. We have studied various functions that can be used for selecting the initial value of the *RD* field, including the following functions:

$$RD(T_{data}, h) = k \cdot T_{data},\ k \geq 1, \quad (6)$$

$$RD(T_{data}, h) = T_{data} \cdot h^n,\ n \geq 0, \quad (7)$$

$$RD(T_{data}, h) = T_{data} \cdot (1+\theta)^h,\ \theta \geq 0 \quad (8)$$

where *k, n* and θ are constant parameters, *h* is the number of hops on the path to be followed, and T_{data} is the burst transmission duration. Alternatively, we can introduce a parameter *h(i)* which is initialized with *h* and decremented adaptively after each hop of the reservation process (in this case we have to recompute the *RD* on every hop).

Depending on the function used, and the choice of the parameters *k*, *n* and θ, different reservation policies can be enforced. For example, in order to preferably serve bursts that traverse a large number of hops (such requests have a lower a priori probability of successfully reserving all the required resources) we can use the function of Eq. (7) with $n>0$, or the function of Eq. (8) with $\theta>0$.

The effect of the choice of the *RD* function and the corresponding parameters on throughput performance

(average overall performance, or performance for bursts that traverse a given number of hops) in classless networks is investigated in part A of section V. The performance of the ERBP protocol is compared to that of other previously proposed schemes in part B of section V. Service differentiation can be provided by mapping different classes of service to different *RD* functions or chossing different parameters, as discussed in part C of section V.

V. Protocol Performance Evaluation

To evaluate the proposed signaling and reservation scheme and compare its performance to that of previously proposed schemes for OBS networks, we developed a discrete-event OBS simulator based on the ns-2 platform [19], [20]. The experiments were conducted assuming the NSFnet backbone network topology [21]. In the NSFnet, all links were assumed to be bi-directional with a single wavelength per direction of bandwidth $C = 40Gb/s$. Propagation delays were taken to be proportional to the physical distance between the nodes, and the message processing times were set equal to $20\mu sec$ per hop. An edge node maintains a separate FIFO for each destination. Bursts arrive at each edge node, according to a Poisson process with rate λ requests/second, and burst destinations are uniformly distributed over all nodes. Burst sizes were assumed to follow an exponential distribution with mean value B, corresponding to mean burst duration equal to $\overline{T_{data}} = B/C$. Typical mean burst sizes and mean burst transmission durations considered in the experiments were B = 10-20$MBytes$ that corresponds to $\overline{T_{data}} = 2-4\,msec$, which are at least one order of magnitude smaller than the mean round trip time of the NSFnet ($\overline{T_{RTT}} = 26\,msec$). Finally, we have set the maximum delay tolerance for all bursts equal to D=0.3sec and the edge node buffer size equal to 256$MBytes$.

Apart from the proposed EBRP scheme, we also simulated (i) a typical Tell-and-Wait (TAW) protocol and (ii) a typical Tell-and-Go (TAG) protocol (namely, the Just-Enough-Time scheme with void filling). For the JET protocol we have considered two different versions that differ in the way they handle contentions. More specifically, in the case of contention, the first version (termed here as JET) simply drops the burst, while the second version employs bursts retransmissions, and thus is termed here as JET-with-retrials. To this end, for all the cases of of EBRP, TAW and JET-with-retrials, if the setup process is blocked at an intermediate node and the burst delay limit D allows, the source re-attempts the reservation by re-transmitting the SETUP message, until either a successful reservation is made or until the delay limit D expires. The FIFO property in every virtual output queue is maintained and the scheduling manager does not proceed to serve the next burst residing in a queue until the previous burst is successfully transmitted or finally rejected.

We used the data loss ratio as the main metric for assessing protocol performance. For the two-way reservation schemes (TAW and EBRP) and the JET-with-retrials, the data loss ratio refers to the size of the bursts lost due to buffer overflow or due to burst time delay expiration at the edge nodes (recall that when a two-way reservation scheme is used, the core network is free of blocking, while in the JET-with-retrials scheme a burst is re-transmitted if it is dropped in the core). In the case of the simple JET protocol, the data loss ratio refers to the size of bursts dropped in the core of the network due to contention. It is worth noting that the *data loss ratio* metric we use differs from the *burst loss ratio,* since it takes into account not only the probability of dropping a burst but also the size of the dropped data, and thus is closely related to the actual throughput performance that can be achieved by the examined protocols. Additional performance metrics measured in our simulations were the average number of SETUP re-transmissions required for a successful reservation and the average end-to-end delay experienced by a burst, defined as the average time that elapses between the time its assembly is completed and the time it reaches its destination.

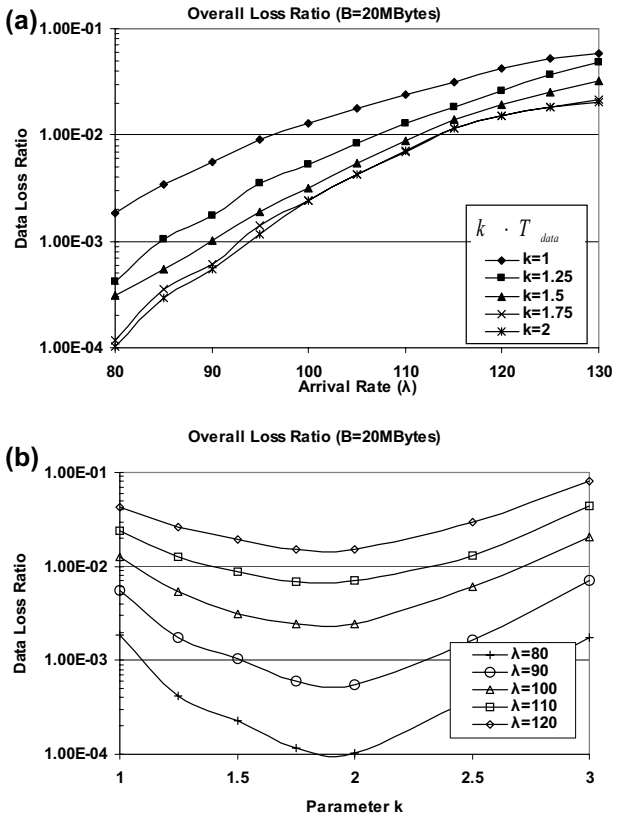

Figure 3: (a) The data loss ratio for $RD(T_{data}, h) = k \cdot T_{data}$ and k=1, 1.25, 1.5, 1.75 and 2. (b) Corresponding data loss ratios for different values of the arrival rate λ versus parameter k.

A. Effect of Reservation Duration (RD) function

In this section, we investigate several choices of the initial value of the Reservation Duration (*RD*) field of the SETUP packet. The *RD* field is initialized as a dynamically varying parameter on a per burst basis using the three functions presented in Equations (6), (7) and (8) of Section IV. Thus, the initial value of the *RD* field is selected based on the burst duration, the number of hops to the destination, and the values

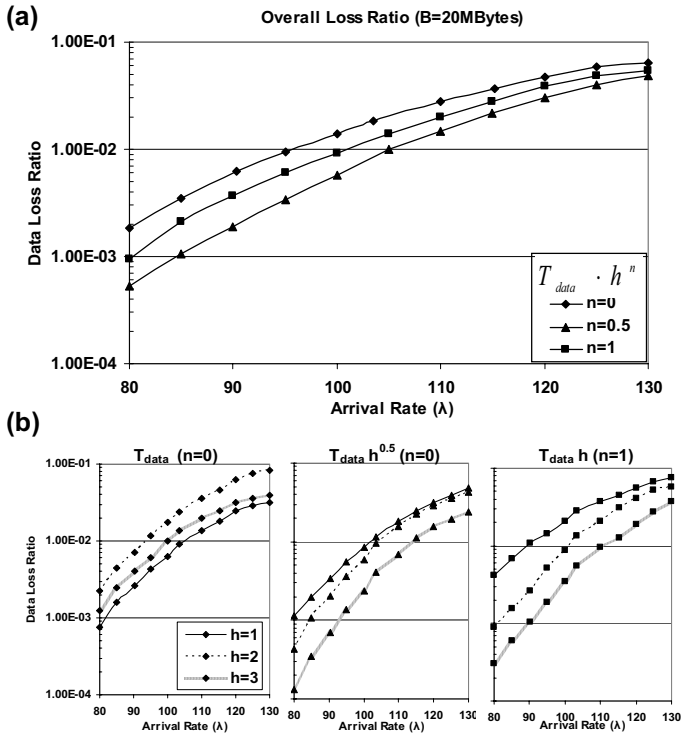

Figure 4: (a) Overall data loss ratio for $RD(T_{data}, h) = T_{data} \cdot h^n$ and $n = 0$, 0.5, and 1. (b) Corresponding data loss ratios for bursts traversing h=1, h=2 and h=3 hops, for different choices of the parameter n (initializing the *RD* field according to $RD = T_{data}$, $RD = T_{data} \cdot \sqrt{h}$ and $RD = T_{data} \cdot h$).

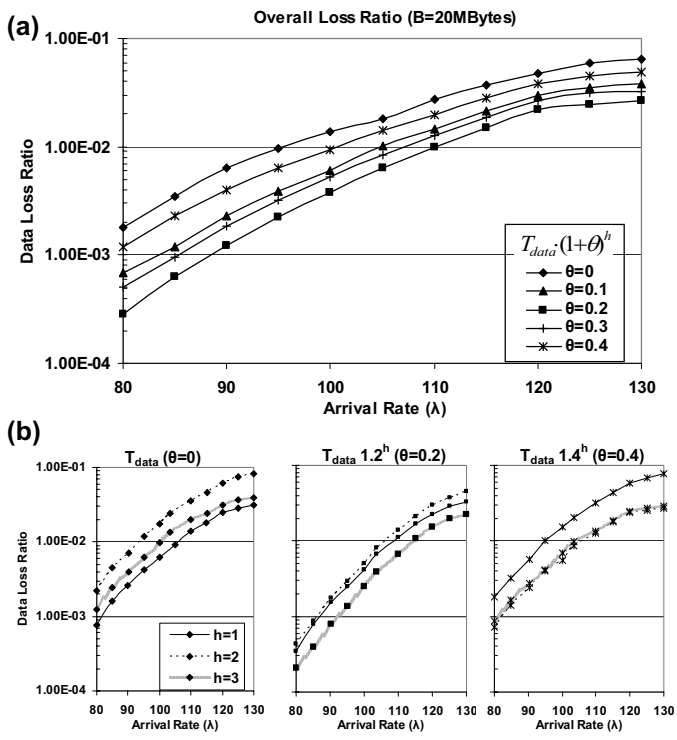

Figure 5: (a) Overall data loss ratio for $RD(T_{data}, h) = T_{data}(1+\theta)^h$ and $\theta = 0$, 0.1, 0.2, 0.3 and 0.4. (b) Corresponding data loss ratios for bursts traversing h=1, h=2 and h=3 hops.

of the parameters k, n and θ. The results presented in this section were obtained for mean burst sizes B=20MBytes, but the performance graphs were similar for other choices of the burst size. We experimented mainly with relatively small values of the parameters k, n and θ, since larger values of these parameters were found to result in worse performance, due to the wasteful reservation of excessive resources during the setup phase.

Figure 3 shows the results obtained when the function $RD(T_{data}, h) = k \cdot T_{data}$ (Eq. (6)) is used for initializing the *RD* field. In particular, Figure 3 (a) shows the data loss ratio in the network for k=1, 1.25, 1.5, 1.75 and 2, as a function of the arrival rate λ per node. Figure 3 (b) shows the data loss ratio versus parameter k for different values of the arrival rate λ. From Figure 3 (a) we observe that the EBRP protocol outperforms a typical two-way scheme that employs delayed reservations (k=1), through the use of its *relaxed delayed reservation* mechanism where the *RD* field is initialized with values larger than T_{data}. Regarding the choice of the parameter k, it can be seen from Figure 3 (b) that all curves exhibit a similar pattern: the loss ratio first decreases as k increases up to a point, and then starts to increase, indicating that beyond that point reserving excess resources has a counter-effect on the acceptance probability of future requests. The optimum performance for the NSFnet topology was observed for values of k close to 2.

Figure 4 (a) shows the results obtained when the function $RD(T_{data}, h) = T_{data} \cdot h^n$ (Eq. (7)) is used for initializing the *RD* field. The idea here is that bursts that travel a large number of hops, and therefore have a smaller a priori probability of successfully reserving the resources they need, should be given more flexibility during the reservation process than bursts that travel a small number of hops. Large values of the parameter n give heavier dependence of the reservation duration *RD* on the number of hops h. The best performance was observed for the case n=0.5, followed by the case n=1, while the worst performance was observed for n=0. These results demonstrate the performance benefits that can be obtained by considering the number of hops h in the initialization of the *RD* field.

The loss ratio shown in Figure 4 (a) corresponds to the overall average data loss ratio in the network, independently of the number of hops on the paths taken. Figure 4 (b) shows the detailed data loss ratios for bursts having to traverse h=1, h=2 and h=3 hops. Clearly, for n=0.5 (middle figure) the loss ratios of the bursts that traverse h=1, 2, or 3 hops are better than the corresponding loss ratios for the cases n=0 (left figure) and n=1 (right figure). As expected, when n=0, bursts having to traverse h=3 hops have a higher data loss ratio than bursts that traverse only 1 or 2 hops. When however n=0.5, traffic that uses paths with h=3 hops exhibits better performance than traffic that uses single hop paths. Therefore, setting n>0 introduces a certain amount of fairness with respect to the traffic destination. Overall our results show that the data loss ratio in the NSFnet, when a function of the form given in Eq. (7) is used, is optimized when

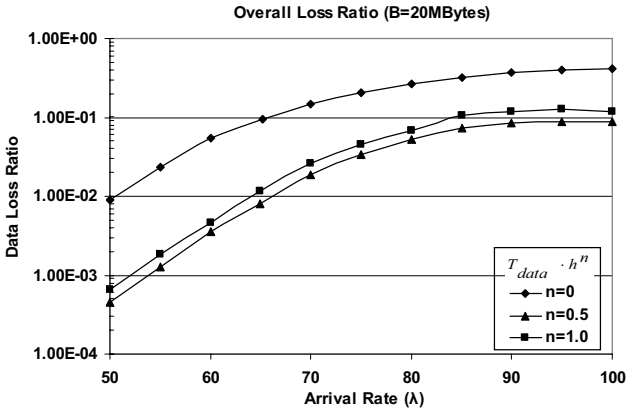

Figure 6: Overall data loss ratio on a 6x6 mesh network for $RD(T_{data},h) = T_{data} \cdot h^n$ and n = 0, 0.5, and 1.

$RD(T_{data},h) = T_{data} \cdot \sqrt{h}$. When the RD field is chosen in this way, the overall data loss performance is improved while at the same time the traffic is treated in a more equitable way, independently to a large degree of its destination and the number of hops it has to traverse.

Figure 5 shows the results obtained when the function $RD(T_{data},h) = T_{data}(1+\theta)^h$ (Eq. (8)) is used for initializing the *RD* field. From Figure 5 it can be seen that the overall data loss ratio exhibits a minimum for θ=*0.2,* while in this case the performance is also more equitable with respect to the burst destination, since bursts that travel a large number of hops are treated more favorably at each hop than bursts that travel a small number of hops. For values of θ greater than 0.2 the performance starts to deteriorate indicating that for such large values of θ the setup packet reserves excessive resources, having a counter-effect on the acceptance probability of future requests.

The NSFnet topology, for which the performance results presented so far were obtained, has diameter equal to 3 and average shortest path distance equal to 2.2. To see if the performance results obtained for the NSFnet topology are representative of those that would be obtained for other network topologies, we also experimented extensively with a 6x6 mesh topology. In the 6x6 mesh, the nodes were arranged along a two-dimensional grid topology, with neighboring nodes placed at a distance of 300 km from each other. The traffic and protocol parameters were kept the same as in the previous experiments. The results obtained for the mesh topology were qualitatively similar to those obtained for the NSFnet topology. For example, Figure 6 shows the results obtained for the 6x6 mesh when $RD(T_{data},h) = T_{data} \cdot h^n, n \geq 0$ (Eq. (7)). As was also the case with the NSFnet topology, we observe that the choice $RD(T_{data},h) = T_{data} \cdot \sqrt{h}$ yields again the lowest overall data loss ratio, while treating at the same time bursts that travel over paths of different lengths in a more fair way. However, we can observe that for the 6x6 mesh topology the performance of the case n=1 converges to that of the case n=0.5. Concluding, even though the network topology obviously plays a role in protocol performance, the qualitative conclusions obtained in this section regarding the best choice of the reservation duration (*RD*) parameter of the EBRP protocol, are to a large extent valid for different topologies.

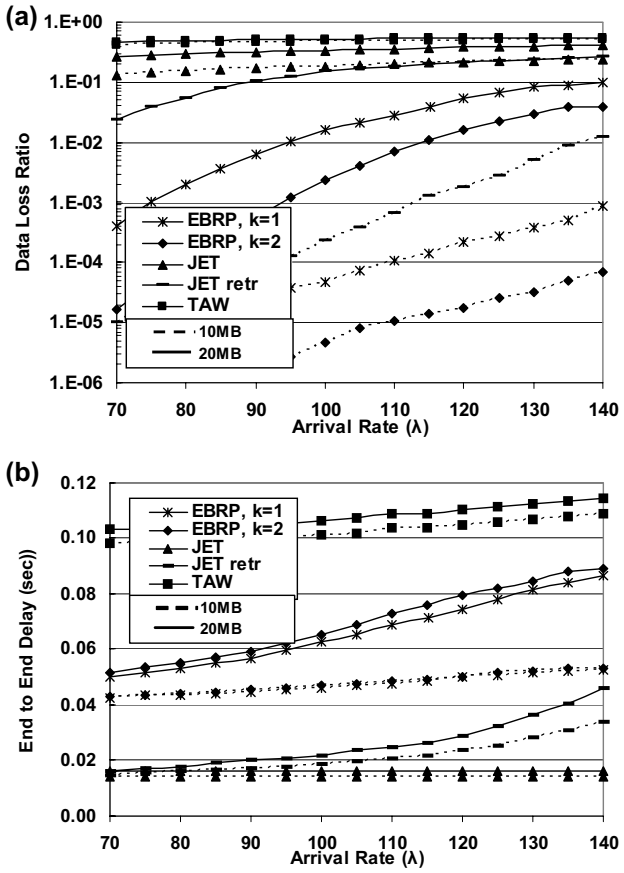

Figure 7: (a) Data loss ratio and (b) End to End Delay of the EBPP protocol with relaxed reservations (case k=2), the EBPP protocol without relaxed reservations (case k=1), the JET protocol, the JET-with-retrials protocol, and the TAW protocol.

B. EBRP performance comparison and assessment

In this section, we compare the performance of the EBRP protocol to that of two other signaling schemes, namely, a typical Tell-and-Wait (TAW) protocol in which resources are reserved upon the reception of the SETUP packet at an intermediate node, and a typical one-way scheme and more specifically the JET with void filling protocol. For the JET protocol we have considered two different versions: in the case of a contention, the first version (termed as JET) simply discards the burst in the core, while the second version (termed JET-with-retrials) uses reject packets in order to enable burst retransmissions.

In the experiments carried out in this section for the EBRP protocol, the *RD* field was initialized as $RD(T_{data}) = 2 \cdot T_{data}$ (i.e. we used Eq. (6) with k=2). Figures 7 (a) and (b) show the data loss ratios and the average end-to-end delays of the EBRP, JET, JET-with-retrials, and TAW schemes for mean burst sizes B=10 and 20Mbytes. For comparison purposes, we also include in Figure 7 the EBRP performance for the case

$RD(T_{data}, h) = T_{data}$ (i.e., Eq. (6) with k=1), where the *relaxed delayed reservation* mechanism is not used. Regarding the data loss ratio metric illustrated in Figure 7 (a), we observe that the EBRP protocol outperforms the JET, JET-with-retrials, and TAW protocols for a wide range of λ values. It is worth noting that the improvement in the performance of the EBRP protocol *with* relaxed delayed reservations (case k=2) is better than that of the EBRP protocol *without* relaxed delayed reservations by more than one order of magnitude.

As expected, the performance of the JET and TAW schemes was not satisfactory: In the former scheme, the strict delay requirements in the setup process combined with the support of a single wavelength per link (no wavelength conversion to support contention resolution) yield high loss ratios in the core. The performance is improved when burst retransmission is enabled (JET-with-retrials). In the latter scheme, the reservation of resources at a node immediately upon the reception of the SETUP packet leads to wasteful utilization of the capacity and low acceptance probability for future connections.

With respect to the end-to-end delay (Figure 7 (b)), the EBRP protocol performs better than the TAW scheme but worse than the JET and JET-with-retrials schemes, for the illustrated arrival rates λ. In the JET scheme, bursts are transmitted almost immediately after the completion of the assembly process (of course, they have to wait for the transmission of the previous burst and the time offset required for setup), and thus the end-to-end delay is mainly determined by the propagation delay. Since the number of contentions (and thus the number of retrials) increases as λ increases, in the case of the JET-with-retrials scheme the end-to-end delay performance deteriorates as λ increases. On the other hand, in the two-way schemes, the end-to-end delay is the sum of the propagation delay plus the round-trip time of the two-way reservation process. For heavy load, however, we have to take into account the additional delay required for setup retrials in case the first reservation attempt is not successful. The difference between the EBRP and the TAW scheme lies mainly in the better utilization of resources accomplished by the former and thus in the smaller number of setup retrials required until a successful reservation. Finally, we observe that the effect of the *relaxed delayed reservation* mechanism (EBRP with k=2 as opposed to k=1) on the end-to-end delay is negligible, indicating that the improvement in the data loss ratio obtained by using *relaxed* reservations comes at little cost in terms of delay.

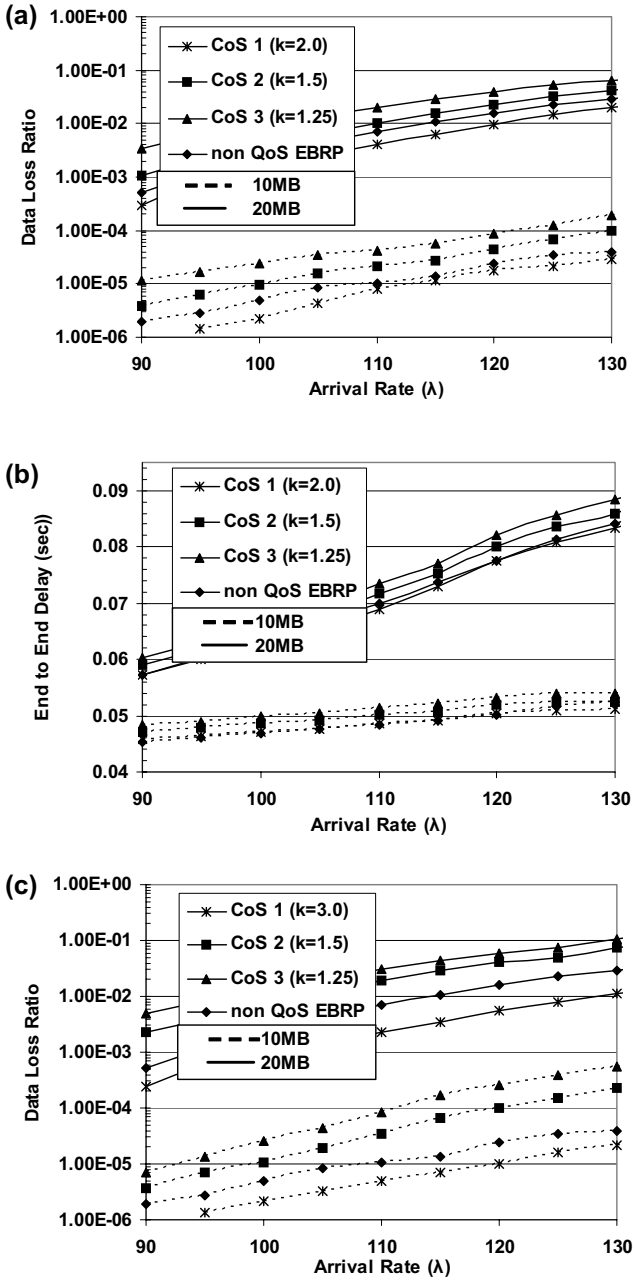

Figure 8: (a) Data loss ratio, and (b) end-to-end delay for three different classes of service that are evenly generated and whose *RD* fields are initialized with different functions. In particular, $RD = k_i \cdot T_{data}$ and Classes 1 to 3 corresponds to k_i=2, 1.5 and 1.25 respectively. (c) Data loss ratio when $RD = k_i \cdot T_{data}$ and Classes 1 to 3 corresponds to k_i=3, 1.5 and 1.25 respectively.

C. Case of multiple classes of service

In this section we investigate the performance of the EBRP protocol when multiple classes of services are employed. In particular we assume that the edge nodes maintain a set of virtual queues for each destination node, each corresponding to a different Forwarding Equivalence Class (FEC) as in [2]. Each FEC is assigned a different priority class i and initializes its *RD* field according to

$$RD_i(T_{data}, h) = k_i \cdot T_{data},$$

where the parameters k_i are used to differentiate the QoS experienced by each class i. Assuming that FEC *i-1* has higher priority than FEC i, we have $k_{i-1} > k_i$.

In our experiments, we assumed three Classes of Service (CoS) with parameters k_1 = 2, k_2 = 1.5, k_3 =1.25. Bursts are generated according to a Poisson process with rate λ requests per second, and belong to equal probability to each of the three classes. Figures 8 (a) and (b) shows the data loss ratio

and average end-to-end delay for each class. From both figures, it is clear that high-priority traffic (CoS-1) exhibit the lowest data loss ratio and the lowest delay.

Note that in the case of the EBRP protocol, the excess resources reserved by a class can negatively affect the performance of the other classes. This was verified by assigning a higher value to the parameter k_1 of class CoS-1, and keeping the same the parameters of the other classes. Figure 8 (c) shows the corresponding data loss ratios for all classes when $k_1 = 3$, $k_2 = 1.5$, $k_3 = 1.25$. It is clear that the improvement in the data loss ratio of CoS-1 comes at the expense of a performance degradation of classes CoS-2 and CoS-3.

VI. Conclusions

In this paper we have presented the Efficient Burst Reservation Protocol (EBRP), a new two-way burst signaling and reservation scheme suitable for bufferless OBS networks. The protocol employs two mechanisms: (i) *relaxed delayed reservations* to increase the burst forwarding probability and provide service differentiation and (ii) *in-advance reservations* for scheduling bursts for future time intervals when capacity is not available for them at the time it is requested, so as to decrease the overhead associated with the repetition of the call setup phase. A key parameter of the EBRP protocol is the *reservation duration* (*RD*), which is initialized as a dynamic per burst parameter, and provides control over the allowed degree of the delayed reservation mechanism. We have studied various choices for initializing the *RD* field as a function of the burst duration and the number of hops to the destination, and we have shown ways for providing service differentiation using different *RD* functions for different classes of service. The performance evaluation results show that the EBRP protocol with suitably chosen parameters outperforms other delayed reservation protocols in terms of data loss ratio and resource utilization for bursts that can tolerate the round trip delay required by the two-way reservation process.

VII. Acknowledgements

This work has been supported by European Commission through the project IST-LASAGNE, the Greek General Secretariat for Research and Technology -GSRT- via the PENED and the Operational Programme for Education and Initial Vocational Training -EPEAEK- via the PYTHAGORAS project.

References

[1] C. Qiao and M. Yoo, “Optical burst switching (OBS)—A new paradigm for an optical internet”, Journal of High Speed Networks, vol. 8, pp. 69–84, 1999.

[2] M. Dueser and P. Bayvel. “Analysis of a dynamically wavelength-routed optical burst switched network architecture”, IEEE/OSA Journal of Lightwave Technology, 20:574-585, April 2002.

[3] M. Dueser and P. Bayvel, “Performance of a dynamically wavelength-routed optical burst switched network”, IEEE Photonics Technology Letters, 14(2):239-241, Feb 2002.

[4] R. Karanam, V. M. Vokkarane, and J. P. Jue, “Intermediate Node Initiated (INI) Signaling: A Hybrid Reservation Technique for Optical Burst-Switched Networks”, Proceedings, IEEE/OSA Optical Fiber Communication Conference 2003, Atlanta, GA, March 2003.

[5] J. Y. Wei and R. I. MacFarland Jr, “Just-In-Time signaling for WDM optical burst switching networks”, IEEE/OSA Journal of Lightwave Technology, vol. 18, pp. 2019–37, Dec. 2000.

[6] E.A. Varvarigos, V. Sharma, “The ready-to-go virtual circuit protocol: a loss-free protocol for multigigabit networks using FIFO buffers”, IEEE/ACM Transactions on Networking, vol.5, (no.5): pp.705-18, Oct. 1997.

[7] M. Yoo and C. Qiao, “Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks”, IEEE/LEOS Technology Global Information Infrastructure, pp. 26–27, Aug. 1997.

[8] J. S. Turner. “Terabit burst switching”, Journal of High Speed Networks, 8 (1):3-16, January 1999.

[9] Y. Xiong, M. Vandenhoute, and H. Cankaya, “Control architecture in optical burst-switched WDM networks”, IEEE Journal on Selected Areas in Communications, vol. 18, pp. 1838–1851, October 2000.

[10] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, “JumpStart: A just-in-time signaling architecture for WDM burst-switched networks”, IEEE Communications, 40(2):82-89, February 2002.

[11] J. Xu, C. Qiao, J. Li, and G. Xu, “Efficient channel scheduling algorithms in optical burst switched networks”, Proceedings of INFOCOMM, 2003, vol. 3, pp. 2268–2278.

[12] C. Gauger, K. Dolzer, J. Spath, and S. Bodamer, “Service differentiation in optical burst switching networks”, Beitrage zur 2.ITG Fachtagung Photonische Netze, pages 124 -132, March 2001.

[13] M. Yoo, C. Qiao and S. Dixit, “QoS Performance of Optical Burst Switching in IP-over-WDM Networks”, IEEE Journal on Selected Areas in Communications, vol. 18, pp. 2062-2071, October 2000.

[14] V. Vokkarane, and, J.P. Jue, “Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks”, IEEE Journal on Selected Areas in Communications., vol. 21, no. 7, pp.1198-1209, Sept. 2003.

[15] W. Liao and C. Loi, “Providing service differentiation for optical-burst-switched networks”, IEEE/OSA Journal of Lightwave Technology., vol. 22, no. 7, pp. 1651 – 1660, July 2004.

[16] Q. Zhang, V.M. Vokkarane, J.P. Jue, and Biao Chen, “Absolute QoS Differentiation in Optical Burst-Switched Networks”, IEEE Journal on Selected Areas in Communications, vol. 22, no. 9, pp. 1781-1795, Nov. 2004.

[17] J. Liu, N. Ansari, and T.J. Ott, “FRR for Latency Reduction and QoS Provisioning in OBS Networks”, IEEE Journal on Selected Areas in Communications, vol. 21, no. 7, pp. 1210-1219, Sept. 2003.

[18] E.A. Varvarigos and V. Sharma, “An efficient reservation connection control protocol for gigabit networks”, Computer Networks and ISDN Systems 30, pp. 1135–1156, 1998.

[19] Optical WDM Network Simulator (OWns), University of Maryland DAWN research lab. http://dawn.cs.umbc.edu/owns/

[20] Bo Wen, Nilesh M. Bhide, Ramakrishna K. Shenai, and Krishna M. Sivalingam, “Optical Wavelength Division Multiplexing (WDM) Network Simulator (OWns): Architecture and Performance Studies”, SPIE Optical Networks Magazine Special Issue on “Simulation, CAD, and Measurement of Optical Networks”, March 2001.

[21] http://moat.nlanr.net/INFRA/NSFNET.html

[22] J.Li, G. Mohan and K.C. Chua, “Dynamic load balancing in IP-over-WDM optical burst switching networks”, Computer Networks, vol 47, pp. 393-408.

[23] Guoping Zeng, Kejie Lu, and Imrich Chlamtac, "On the Conservation Law in Optical Burst Switching Networks", Procedings Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECT) 2004, pp.124--129, San Jose, CA, July, 2004.