Seventh International Conference on Networking

Fast Reservation Protocols for Latency Reduction in Optical Burst-Switched Networks Based on Predictions

Kyriaki Seklou and Emmanouel Varvarigos
Department of Computer Engineering and Informatics, University of Patras and
Research Academic Computer Technology Institute, Patras, Greece.

Abstract— **We propose and evaluate fast reservation (FR) protocols for Optical Burst Switched (OBS) networks. The proposed reservation schemes aim at reducing the end-to-end delay of a data burst, by sending the Burst Header Packet (BHP) in the core network before the burst assembly is completed at the ingress node. We use linear prediction filters to estimate the expected length of the burst and the time needed for the burstification process to complete. A BHP packet carrying these estimates is sent before burst completion, in order to reserve bandwidth at each intermediate node for the time interval the burst is expected to pass from that node. Reducing the total time needed for a packet to be transported over an OBS network is important, especially for real-time applications. Reserving bandwidth only for the time interval it is actual going to be used by a burst is important for network utilization efficiency. In the simulations conducted we evaluate the proposed extensions and prove their usefulness.**

I. INTRODUCTION

Optical Burst Switching (OBS) [1] has been proposed as a way to simplify network control and switching hardware in the future generation optical Internet. OBS combines the best attributes of optical circuit-switching and optical packet switching. Circuit switching is not efficient for bursty traffic and requires the aggregation of microflows into circuits, meaning that fine granularity and control over the QoS of individual microflows are lost. All-optical packet switching, on the other hand, is still considered to be at its infancy since the technology required to implement it is not yet mature.

The basic ideas underlying an OBS system are twofold: the assembly of data packets into bursts that are switched using a single label, and the decoupling of the transmission of the control header from the transmission of the data payload. During the burstification process, multiple packets are aggregated into big containers (data bursts) at the network ingress. Typically, an edge router maintains a separate (virtual) queue for each Forwarding Equivalent Class (FEC), defined by the destination and Quality of Service (QoS) parameters. A control header, also called a burst header packet (BHP), is transmitted at an earlier time than the data payload in order to reserve the required bandwidth and configure the switches along the path. After a short time offset (TO) from the BHP transmission, the data burst is released and switched throughout the network all-optically. The separation between control and data maintains data transparency and leads to a better synergy of mature electronic technologies (which process the BHP) and advanced optical technologies (which process the data burst).

The end-to-end delay over an OBS network mainly consists of four components: (i) the burst assembly delay at the edge node, (ii) the path setup delay caused by the BHP, (iii) the burst transmission time, and (iv) the propagation delay of the burst in the core network. The two last delay components depend on the path selected and the available bandwidth on that path and cannot be reduced through clever design of the signaling protocols. Our work focuses on the first two delay components, and uses pipelining techniques to reduce their combined effect and the overall end-to-end delay. The end-to-end delay is a crucial QoS parameter for a number of applications such as voice, videoconferencing and real time applications. The burst dropping probability is another factor of interest that influences QoS.

The burst assembly process starts with the arrival of the first packet at an empty queue and continues until a predefined threshold is reached. Different assembly strategies define differently this threshold. These strategies generally try to balance between two objectives: the burstification delay and the burst size. Short burstification delays and large burst sizes are desirable, in order to reduce the total end-to-end delay and the processing overhead, respectively. These objectives, however, contradict each other since increasing the burst size also increases the burstification delay. When the burstification threshold is reached and the burst has been formed, the BHP is sent to the core network, followed after a short time interval, called time-offset and denoted by t_0, by the burst itself. The time-offset t_0 is a minimum separation between the BHP and the data burst that allows intermediate OBS nodes enough time to configure their switching fabrics for the burst that follows.

A number of burst assembly algorithms have been proposed in the literature, including the time-based algorithm (abbreviated T_{MAX} algorithm) and the length-based algorithm (abbreviated BS_{MIN} algorithm) [2][3][4]. In the T_{MAX} algorithm, a time counter is started upon the arrival of a packet at an empty FEC queue. When the counter reaches the threshold T_{MAX}, a burst is created, and is queued for transmission on the data channel. Next its BHP is sent and the actual data burst is sent after the corresponding offset time. The time counter is then reset to zero and it remains so until the next packet arrival at the queue. In the BS_{MIN} algorithm, the threshold specifies the number of packets to be aggregated into a burst, or the size of the burst in bytes if the packets are of variable size. Once the threshold is reached, the burst is created, its BHP is sent into the optical network and the data burst is transmitted after an appropriate offset time. Hybrid schemes have also been proposed [6], where a burst is

978-0-7695-3106-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICN.2008.90

completed when either the time threshold T_{MAX} or the burst length threshold BS_{MIN} is reached, whichever happens first. An alternative recently proposed burst assembly algorithm is the average delay-based algorithm (abbreviated T_{AVE} algorithm) [5], where a new burst is constructed when the average delay of the packets that comprise it reach a predefined threshold. This method guarantees a desired target value for the average delay of the packets comprising a burst and tends to minimize packet delay jitter, which is particularly important for TCP performance.

In this work we propose and evaluate a number of fast reservation (FR) schemes that can be combined with the BS_{MIN}, T_{MAX} and T_{AVE} burst assembly algorithms. These extensions use one or two linear prediction filters, respectively, to estimate the length of the burst and/or the time needed for the burstification process to complete. In contrast to standard OBS signaling protocols, in our work the BHP is sent to the core network before the burst assembly process is completed, to reserve the appropriate resources. To do so, it uses the estimated values of the burst length and assembly completion time, instead of the actual values that are not yet known at the time BHP is sent, in order to reserve bandwidth at each intermediate node for the interval the burst is expected to pass from that node. Estimating the length of the burst is required in order to reserve the required resources in the core network for the right duration for the burst's all-optical transmission. Estimating the duration of the burst assembly process is required in order to determine the time these reservations should start at the core nodes. Our goal is (i) to reduce the end-to-end delay of a data burst, by minimizing the burst pretransmission time, while (ii) using bandwidth efficiently by reserving resources for a time duration that is close to the minimum possible.

The applicability of traffic prediction at the ingress nodes of OBS networks has been examined and verified in [8][11][12][13]. In [8] the authors evaluate the use of a linear prediction filter along with the T_{MAX} burst assembly algorithm to reduce the burst pre-transmission delay in a way similar to the present work. Specifically, in [8] the used filter predicts the size of the next data burst and a BHP is sent to the core network before the burst assembly completes. We show that prediction can also be used along with the BS_{MIN} and T_{AVE} algorithms, reducing the end-to-end delay of the bursts.

The remainder of the paper is organized as follows. In Section II we describe the proposed scheme. In Section III we present our simulation results. Finally in Section IV we present our conclusions.

II. PROPOSED SCHEME

A. System Model

Figure 1 highlights the architecture of a typical OBS network, consisting of a cloud of optical core routers, organized as a flat mesh, with edge routers at the edges of the cloud. Core nodes are responsible for forwarding the burst to the proper destination edge node. Edge nodes are of two types: ingress (source) nodes and egress (destination) nodes. A node can be both a source and a destination. Burstification is performed at ingresses, where a burstification control unit (BCU) resides and coordinates the transmission of data and control packets.

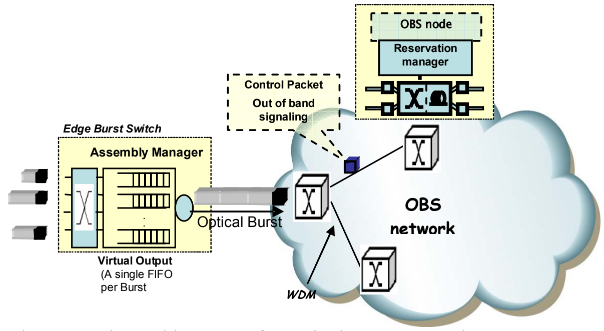

Figure 1: The architecture of a typical OBS network.

Signaling protocols for OBS systems are based on two alternative schemes: "tell-and-wait" (TAW) and "tell-and-go" (TAG). While the former features a two-way reservation process, the latter uses one-way signaling that releases the burst without waiting for the confirmation of the successful establishment of the path. Thus, in TAW schemes reservation requests can be blocked, but bursts are guaranteed to arrive at their destination once they enter the network, while in TAG schemes bursts can be dropped at the core nodes since resources are not reserved for them in advance. In this work we adopt the TAG scheme, since it incurs a smaller pre-transmission delay at the edge (source) node. We evaluate the use of a prediction method in order to further reduce the pre-transmission delay of the data burst (the time offset) required in the TAG scheme. For the resource reservations in the core network, we use the "Just Enough Time" (JET) protocol which employs delayed reservation (DR) [7]. In JET, the resources at intermediate nodes are reserved for the incoming burst starting from the arrival of the burst at a link until its departure. This approach enables the reservation of resources for the precise burst transmission duration, resulting in efficient bandwidth utilization and high system throughput. An alternative TAG scheme is the "Just In Time" (JIT) protocol. JIT does not employ delayed reservations and thus the bandwidth is reserved starting at the time the BHP arrives at a link.

B. Fast Reservation Protocols

Our scheme is inspired from the scheme presented in [8] that uses a prediction filter to propose a fast reservation protocol that can be combined with the T_{MAX} assembly algorithm. The scheme in [8] was also implemented in the present work so as to compare it to our corresponding results for the BS_{MIN} and T_{AVE} assembly algorithms.

Let $L(k)$ be the k^{th} burst's size (in bits) and $D(k)$ be its assembly process duration (Figure 2). If both were known at the beginning of a burst assembly period, we could start the reservation process at that time, reducing the overall delay. Since $L(k)$ and $D(k)$ are not known in advance, the idea is to start the reservation process using estimates of these variables. In the time-based burst assembly algorithm $D(k)$ is fixed and equal to T_{MAX} (therefore, we only have to estimate $L(k)$), while in the length-based algorithm $L(k)$ is fixed and equal to BS_{MIN} (therefore, we then only have to estimate $D(k)$). In the average delay-based algorithm both the burst length $L(k)$ and the assembly duration $D(k)$ vary and have to be estimated.

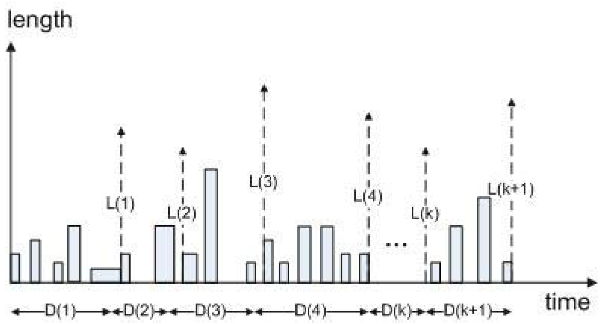

Figure 2: Prediction is performed based on the k previous burst lengths and assembly durations.

The fast reservation (FR) protocol for the T_{AVE} assembly algorithm is illustrated in Figure 3 (the cases of the T_{MAX} algorithm and the BS_{MIN} algorithm are simpler): Upon the beginning of a new burst assembly period we use two Least Mean Squares (LMS) filters [9] to predict burst related values. Using these predictions we can send, at the beginning of the burst assembly process, a BHP to reserve in advance the necessary resources, instead of waiting for the burst assembly process to complete.

Specifically, the first LMS filter is used to obtain a prediction of the length $\hat{L}(k)$ (in bits) of the k^{th} burst to be formed; this value is included in the BHP and is used to reserve bandwidth for the right (if the prediction is accurate) time duration. The second filter is used to obtain a prediction of the assembly process duration $\hat{D}(k)$; this value is also included in the BHP and is used to reserve bandwidth at each intermediate link starting at the right (if the prediction is accurate) time instant. To reduce the effects of prediction errors in the burst length, we add a small margin δ in the estimated burst length $\hat{L}(k)$, in order to reduce the probability that bandwidth is reserved for less time than the actual burst duration. No matter how accurate is the filter, the actual length $L(k)$ of the k^{th} burst will be larger than $\hat{L}(k)$ with probability that is typically close to 50%; however, $L(k)$ will be smaller than $\hat{L}(k)+\delta$ with high probability provided that the prediction filter is good and δ is large enough. Similarly, to reduce the effects of prediction errors in the assembly process duration, we subtract a small margin ε from the estimated duration $\hat{D}(k)$ of the burst assembly process. This is because $\hat{D}(k)$ is used to calculate the time at which reservations at intermediate links should start, and in case of uncertainty, it is safer to start reservations a little earlier than the predicted starting time. By using these margins, the reservation starts earlier than the expected time by ε and finishes later than the expected time by $\varepsilon + \delta/C$, where C is the reserved bandwidth, so that we can be reasonably certain that the burst will find capacity already reserved for it when it arrives at a node. Therefore, bandwidth is reserved at each intermediate node for the time period:

$$\left[\hat{D}(k)-\varepsilon, \hat{D}(k)+\varepsilon+\frac{\hat{L}(k)+\delta}{C} \right], \tag{1}$$

where times are relative to the arrival time of the BHP at each node. Note that the estimators of $L(k)$ and $D(k)$ are unbiased, and capacity is reserved for a burst for time $2\varepsilon+\delta/C$ more on the average than the minimum required. The inefficiency caused by this is negligible if ε and δ are small.

When burst assembly is completed, the predicted values $\hat{D}(k)$ and $\hat{L}(k)$ are compared with the real values $D(k)$ and $L(k)$. The ingress node sends the burst after a small (pre-transmission) interval t_x, calculated so as to compensate for predictions errors, as will be described shortly. If the burst is sent after time t_x, the time period the burst actually traverses the network is:

$$\left[D(k)+t_x, D(k)+t_x+\frac{L(k)}{C} \right], \tag{2}$$

where $L(k)$ is the burst's actual length and $D(k)$ its assembly duration.

In order for the in advance reservation to be successful, the reservation time period must contain the burst's actual transmission period. That is, the reservation at any core node should start before the burst arrives and should finish after the burst's departure. So, based on (1) and (2) the following conditions must hold:

$$t_x + D(k) > \hat{D}(k)-\varepsilon, \tag{3a}$$

and

$$t_x + D(k) + \frac{L(k)}{C} < \hat{D}(k)+\varepsilon+\frac{\hat{L}(k)+\delta}{C}. \tag{3b}$$

The pretransmission time t_x is chosen equal to

$$t_x = \max(\hat{D}(k)-\varepsilon-D(k), 0),$$

so as to minimize pre-transmission delay, while always satisfying (3a).Thus,

$$\Pr(t_x = 0) = \Pr(D(k) > \hat{D}(k)-\varepsilon),$$

and the pre-transmission delay will be 0 with high probability. A sufficient set of conditions to satisfy (3b) is

$$L(k) < \hat{L}(k)+\delta, \tag{4a}$$

and

$$D(k) < \hat{D}(k)+\varepsilon, \tag{4b}$$

which will be both valid with high probability. Thus, if the predictor is good, the reservation will be successful and the pre-transmission delay will be zero with high probability.

If the conditions (3a) and (3b) cannot be simultaneously satisfied for any choice of t_x, the transmitted BHP is a failure and we have to transmit a new BHP to cancel the old reservation and perform a new one with the actual burst size $L(k)$ and the right reservation starting time.

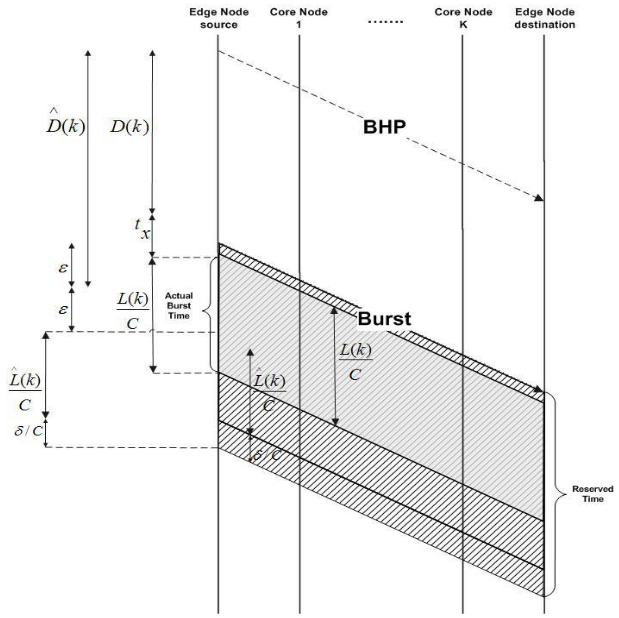

Figure 3: A successful reservation for the T_{AVE} burst assembly algorithm, using two predictive filters. One filter predicts the burst length $\hat{L}(k)$ and the other the burst assembly duration $\hat{D}(k)$.

In the case of the BS_{MIN} assembly algorithm, the length *L(k)* of the burst is fixed and known a priori, since it constitutes the assembly threshold. In that case reservations are performed as in Figure 3 but with $L(k) = \mathrm{BS_{MIN}}$ and $\delta = 0$. A filter is used to obtain the estimated value of the k^{th} burst assembly duration $\hat{D}(k)$, on which we add a small margin of correction ε to compensate for the case the prediction turns out to be larger or smaller than the actual value. The BHP is sent to reserve the necessary resources starting a little earlier than the predicted time, without waiting for the burst assembly to complete. Specifically, the BHP, upon its arrival at a core node, reserves bandwidth *C* for time BS_{MIN}/C, starting at time $\hat{D}(k) - \varepsilon$ and finishing at time $\hat{D}(k) + \frac{\mathrm{BS_{MIN}}}{C} + \varepsilon$, relative to its arrival time at that node. When the burst assembly is completed, the actual assembly duration *D(k)* is compared to $\hat{D}(k) - \varepsilon$. This comparison is performed in order to ensure that the reservation of the resources in the network starts at the right time. The pretransmission time is again chosen according to

$$t_x = \max(\hat{D}(k) - \varepsilon - D(k), 0).$$

If

$$D(k) < \hat{D}(k) + \varepsilon, \tag{5}$$

the reservation made by the BHP is successful (if $\ddot{D}(k) - \varepsilon < D(k) < \ddot{D}(k) + \varepsilon$, we additionally have $t_x = 0$). Otherwise, it is a failure and we have to transmit a new BHP to cancel the old reservation and perform a new one.

In the case of the T_{MAX} assembly algorithm, the burst assembly duration *D(k)*= T_{MAX} is a priori known, since it is used as the assembly threshold. In that case reservations are performed as in Figure 3 but with $\hat{D}(k) = \mathrm{T_{MAX}}$ and $\varepsilon = 0$. A filter is used to predict the k^{th} burst length $\hat{L}(k)$, and bandwidth is reserved for time $(\hat{L}(k) + \delta)/C$. When the time threshold T_{MAX} is reached, the burst assembly is completed, and the actual burst length *L(k)* is compared with the predicted length. If

$$\hat{L}(k) + \delta > L(k), \tag{6}$$

the pretransmitted BHP has reserved capacity for enough duration and the reservation is successful. Otherwise, the pretransmitted BHP is considered to be a failure and a new BHP must be sent to cancel the old reservation and perform a new one for the actual burst size *L(k)*.

The transmission of the BHP has to precede the transmission of the burst by at least a time offset equal to t_0, where t_0 is a parameter chosen to account for the extra processing delays the BHP (which is processed electronically) encounters at intermediate nodes when compared with the processing delays encountered by the burst (which is switched all-optically). For example, if t_{el} is the amount of time it takes for a core node to process electronically the BHP and t_{ao} is the amount of time it takes for the core node to configure its switch fabric to set up a connection from an input port to an output port, we can choose

$$t_0 = h \cdot t_{el} + t_{ao}, \tag{7}$$

where *h* is the number of hops on the path.

If the estimate $\hat{D}(k)$ in the length- or average delay-based burst assembly algorithm is less than t_0, the estimate $\hat{D}(k)$ carried by the BHP in the signaling protocol is replaced by $\max(\hat{D}(k), t_0)$. The total burstification and pre-transmission delay when a fast TAG reservation (FR) protocol is used is

$$T_{FR} = \max(t_0, D + t_x) = \max(t_0, \hat{D} - \varepsilon),$$

[for the T_{MAX} algorithm it is max (T_{MAX}, t_0)], while if a standard TAG reservation (SR) protocol is used, it is

$$T_{SR} = D + t_0.$$

Comparing these two expressions, the delay reduction achieved by the pipelining effect of the proposed FR protocol becomes evident.

It is natural to assume that the parameters T_{MAX}, BS_{MIN} and T_{AVE} in the corresponding burst assembly algorithms are chosen so that the average burst assembly duration satisfies $E(D) > t_0$. For example, in the time-based algorithm, it is natural to choose $T_{MAX} > t_0$, since otherwise we could extent the burst assembly period to get larger bursts without any cost in delay. So under these assumptions we can see that the total end-to-end delay can be reduced by about the time offset t_0.

C. Prediction Method

For our predictions we use a Least Mean Squares (LMS) filter, described in [9] and also used for traffic prediction in [8][12]. For a prediction method to be practical in an OBS system, it should not only be accurate, but also simple and fast.

The estimate of the length of burst k is obtained based on the lengths of the previous N bursts according to:

$$\hat{L}(k)=\sum_{i=1}^{N} h(i)\cdot L(k-i)\,, \tag{8}$$

where $h(i)$, i=1,2,...,N, are the coefficients of the predictive filter and N is its length. In our notation $L(k-i)$ is the duration of burst k-i and $\hat{L}(k)$ the predicted length of burst k.

There are a variety of ways to obtain the filter coefficients. In our experiments we used the N-order LMS-based recursive LPF that updates the filter coefficients using a simple and efficient algorithm. Specifically, the coefficients for the k^{th} prediction period are defined as

$$h(k)=h(k\text{-}1)+\mu\cdot e_L(k-1)\cdot L(k\text{-}i)\,, \tag{9}$$

where μ is an adjustable parameter of the LPF filter (step) and $e_L(k$-1) is the residual error between the actual and the predicted length of burst k-1. The time complexity for the coefficient calculation of the LMS-based approach is $O(N)$.

A similar filter is used for the prediction of the duration of the k^{th} assembly period, based on the durations of the previous N periods. The equations used are similar to (10) and (11), with the assembly duration $D(k-i)$ of burst k-i, the predicted duration $\hat{D}(k)$ of assembly period k, and the error $e_D(k$-1) between the actual and the predicted duration of assembly period k-1 replacing $L(k-i)$, $\hat{L}(k)$, $e_L(k$-1), respectively.

D. Choice of the safety margins δ and ε

As mentioned in Section II.B, to reduce the effects of prediction errors in the proposed fast reservation (FR) schemes we use the safety margins δ and ε. Specifically, we add a small margin δ in the estimated burst length $\hat{L}(k)$, in order to reduce the probability that bandwidth is reserved for less time than the actual burst duration. We also subtract a small margin ε from the estimated burst assembly duration $\hat{D}(k)$; this is because $\hat{D}(k)$ is used to calculate the times at which reservations start at intermediate links, and in case of uncertainty, it is safer to start reservations a little earlier than the predicted time, in order to be reasonably certain that the burst will find capacity already reserved for it when it arrives at a node.

The values of δ and ε significantly impact the success probability of the BHP reservation (the larger δ and ε are, the larger the probability) and the system costs (the smaller δ and ε are, the smaller the time interval during which capacity is reserved but not used). Thus, δ is chosen to be a multiple of the root mean square (RMS) of the sample residuals of the LPF, that is,

$$\delta=c_\delta\cdot\sqrt{\frac{\sum_{i=1}^{N} e_L^2(k-i+1)}{N}}\,, \tag{10}$$

where c_δ is a small constant (e.g., less than 2 or 3), to be referred to as the burst length correction parameter in the rest of the document, and $e_L(k)$ is the residual error between the actual and the predicted burst length. Similarly, ε is calculated using the corresponding RMS of the residual errors $e_D(k)$ between the actual and the predicted burst assembly durations, and a duration correction parameter constant c_ε, that is,

$$\varepsilon=c_\varepsilon\cdot\sqrt{\frac{\sum_{i=1}^{N} e_D^2(k-i+1)}{N}}\,. \tag{11}$$

III. SIMULATIONS

We performed our simulations using an OBS network simulator [14] based on the ns-2 platform [10], where we also implemented the used LMS filters. Specifically these filters predict the burst lengths for the T_{MAX} algorithm, the burst assembly durations for the BS_{MIN} algorithm and both the burst lengths and burst assembly durations for the T_{AVE} algorithm.

A. Simulation Parameters

In our experiments we use a simple OBS network consisting of two edge and one core nodes. A link's bandwidth per channel is equal to 10 Gbps. The traffic generated follows a Pareto distribution with the characteristics given in Table I.

TABLE I
PARETO ON/OFF TRAFFIC

Mean On Period	0.002 ms
Mean Off Period	0.001 ms
Packet Size	1500 bytes
Rate	1Gbps

In each experiment we change the shape parameter α of the Pareto distribution so as to change the burstiness of the generated traffic (the corresponding Hurst parameter is $H=\frac{3-a}{2}$); specifically, we used the values α = 1.2,1.4,1.6, 1.8.

The LMS filter's characteristics are presented in Table II. A filter with a smaller order could also have been used.

TABLE II
LMS FILTER'S CHARACTERISTICS

Order	16
Step	1E-14 for T_{MAX}
	1E-1 for BS_{MIN}
	1E-14 and 1E-1 for T_{AVE}

We simulated the T_{MAX} assembly algorithm using different values for the parameter T_{MAX}. To obtain comparable results, in our simulations of the BS_{MIN} assembly algorithm, we used corresponding values for the parameter BS_{MIN}. Specifically, average burst assembly durations D are related to average burst lengths L through

$$L=\rho\cdot R\cdot D\,, \tag{12}$$

where R is the rate of the Pareto traffic and ρ its load:

$$\rho=\frac{on_{period}}{on_{period}+off_{period}}\,. \tag{13}$$

Finally, for the average delay–based assembly algorithm the values of the parameter T_{AVE} we used were one half of the corresponding values used for the parameter T_{MAX}. The values of the parameters used are presented in Table III.

TABLE III
T_{MAX}, BS_{MIN}, T_{AVE} PARAMETERS

T_{MAX}	0.006 sec, 0.008 sec, 0.01 sec
BS_{MIN}	488 KB, 651 KB, 813 KB
T_{AVE}	0.003 sec, 0.004 sec, 0.005 sec

In our experiments we use the following performance metrics:

- The relative error of the prediction, defined as the inverse of the signal-to-noise ratio (SNR):

$$SNR^{-1} = \frac{\sum e^2(k)}{\sum L^2(k)}. \tag{14}$$

  For the T_{MAX} algorithm we have $e_L(k) = L(k) - \hat{L}(k)$, while for the BS_{MIN} algorithm $e_D(k) = D(k) - \hat{D}(k)$.

- The probability of successful reservations. In the T_{MAX} algorithm a reservation is successful if the size of the predicted burst (plus ε) is bigger than the actual burst size, that is, if Eq. (6) holds. In the BS_{MIN} algorithm a reservation is successful if the predicted burst assembly duration (plus δ) is larger than the actual assembly duration, that is, if Eq. (5) holds. Finally in the T_{AVE} algorithm a reservation is considered successful if the conditions (4a) and (4b) are valid.

Generally we want to have a small relative error and a large probability of successful reservations.

B. Simulation Results

In our simulations we evaluated the proposed fast reservation scheme. We first present our results for the T_{MAX} and BS_{MIN} burst assembly algorithms in order to separately evaluate the two LMS filters used. In the end the results for the T_{AVE} algorithm are also presented, where both LMS filters are used.

T_{MAX} algorithm

From Figure 4 we observe that the relative error of the burst length prediction is quite small and decreases as the shape parameter α increases.

Also we evaluated the burst length filter's error and find out that for burst assembly period D = T_{MAX} = 0.01 secs (which corresponds to about 813 KB of burst length) and for shape parameter α = 1.4 the average error measured is equal to 0.135831 KB, which is quite small and is due to statistical reasons.

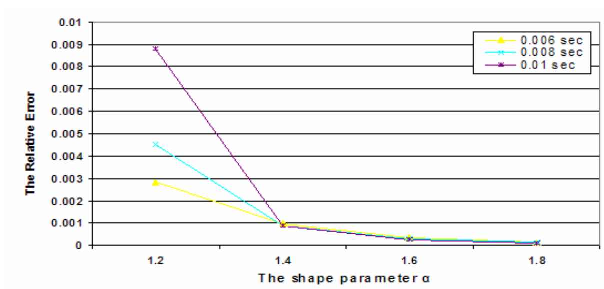

Figure 4: The relative error of the burst length prediction versus the shape parameter α, for three different values of the burst assembly duration D = T_{MAX} for the T_{MAX} assembly algorithm.

The probability of successful reservations was also evaluated for different safety margins δ, using the following correction parameters: c_δ = 0, 1, 2, 3. The experiments were conducted for burst assembly period D = T_{MAX} = 0.006 secs. Figure 5 shows the probability of successful reservations for various values of the shape parameter α and for various correction parameters c_δ. We observe that for c_δ equal to 2 or 3 the results are quite satisfactory, and that the probability of successful predictions increases as c_δ increases. These results are consistent with the results also presented in [8].

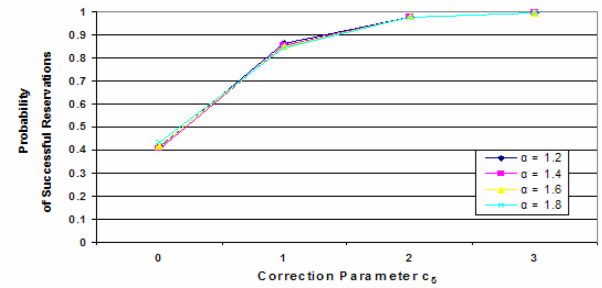

Figure 5: The probability of successful reservations versus the correction parameter c_δ, for different values of the shape parameter α for the T_{MAX} assembly algorithm. The burst assembly period was D = T_{MAX} = 0.006 secs.

BS_{MIN} algorithm

Figure 6 shows the relative error of the prediction for various values of the shape parameter α and for three different burst lengths L = BS_{MIN}. These three values correspond to the values we used for the burst assembly duration D, according to (12). As in the case of the T_{MAX}, we observe that the relative error of the prediction decreases as the shape parameter α increases.

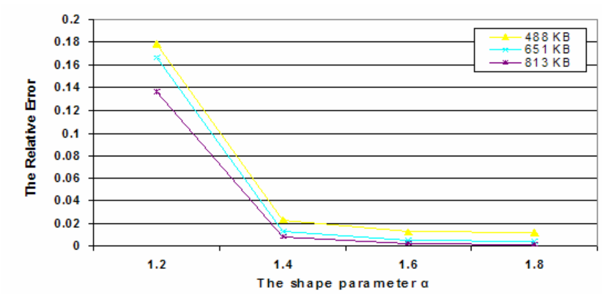

Figure 6: The relative error of the prediction versus the shape parameter α, for three different values of the burst size L = BS_{MIN}, for the BS_{MIN} assembly algorithm.

We also evaluated the prediction error and find that for burst size L = BS_{MIN} = 813 KB (which corresponds to about 0.01 sec burst assembly duration) and for shape parameter α = 1.4 the average error was equal to 0,000194 sec, which is quite small and is due to statistical fluctuations.

The probability of successful reservations of the BS_{MIN} algorithm was evaluated for different safety margins ε, using correction parameters c_ε = 1, 1.02, 1.05, 1.08, 1.1, 1.5, 1, 2, and burst size L = BS_{MIN} = 813 KB. Figure 7 shows the probability of successful reservations for various values of the shape parameter α and correction parameter c_ε. We observe that c_ε = 1.5 or 2 provides sufficiently high success probability and the probability of successful predictions increases as c_ε increases.

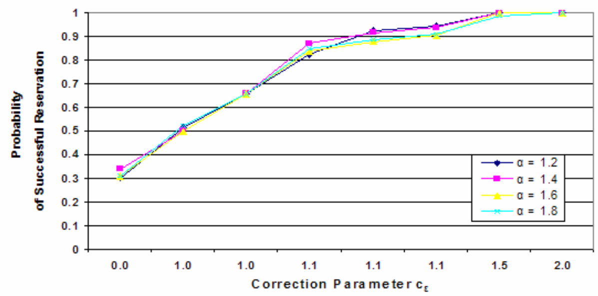

Figure 7: The probability of successful reservations versus the correction parameter c_ε, for four different values of the shape parameter α. The burst size was chosen equal to $L = BS_{MIN} = 813$ KB.

T_{AVE} algorithm

In the T_{AVE} assembly algorithm we use two LMS linear predictive filters in order to predict the length of the next incoming data burst and the burst assembly duration. Figure 8 shows again the probability of successful reservations, using non-zero safety margins δ and ε; in particular, we used a burst length correction parameter $c_\delta = 3$ and an assembly duration correction parameter $c_\varepsilon = 2$. The success probability is calculated by multiplying the probability of successful reservations of the two LMS filters used. The corresponding experiments were conducted for average delay parameter $T_{AVE} = 0.003$ seconds. From Figure 8 we observe that the success probability is quite high and slightly deteriorates when the shape parameter α of the Pareto traffic increases.

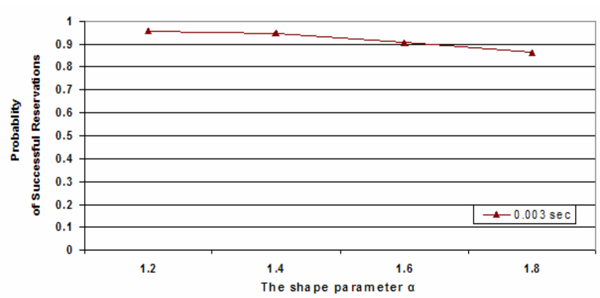

Figure 8: The probability of successful reservations versus the shape parameter α, for the T_{AVE} assembly algorithm and average delay parameter $T_{AVE} = 0.003$ sec. We used length correction parameter $c_\delta = 3$ and an assembly duration correction parameter $c_\varepsilon = 2$.

IV. CONCLUSION

We proposed fast reservation schemes that can be combined with the T_{MAX}, BS_{MIN} and T_{AVE} burst assembly algorithms to reduce the pre-transmission delay in OBS networks, by sending the Burst Header Packet (BHP) in the core network before the burst assembly is completed at the ingress node. We use linear prediction filters to estimate the expected length of the burst and the time needed for the burstification process to complete. These estimates are included in the BHP that is pre-transmitted to make the reservations. We find that the probability of successful reservations in all schemes is very satisfactory, provided that a small correction term (1.5 to 3 times the error RMS) is added to the predicted burst length and burst assembly duration estimates.

REFERENCES

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS) – A new paradigm for an optical internet," J. High Speed Network, Vol. 8, pp. 69–84, 1999.

[2] V. Vokkarane, K. Haridoss, J. Jue, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks", Opticomm, pp. 125-136, 2003.

[3] A. Ge, F. Callegati and L. Tamil, "On Optical Burst Switching and Self-Similar Traffic", IEEE Communications Letters, Vol. 4 (3), 2000.

[4] X. Cao, J. Li, Y. Chen, C. Qiao, "Assembling TCP/IP packets in optical burst switched networks", GLOBECOM, Vol. 3, pp. 2808- 2812, 2002.

[5] K. Christodoulopoulos, E. Varvarigos and K. Vlachos, "A new Burst Assembly Scheme based on the Average Packet Delay and its Performance for TCP Traffic", Optical Switching and Networking, Vol. 4/3-4, Elsevier, pp. 200-212, 2007.

[6] X. Yu, Y. Chen, C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks", Opticomm, 2002.

[7] M.Yoo, C. Qiao, and S. Dixit, "Optical burst switching for service differentiation in the next-generation optical Internet", IEEE Commun. Mag., Vol. 39, pp. 98–104, 2001.

[8] 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 (7), pp. 1210 – 1219, 2003.

[9] J. Treichler, C. Johnson, and M. Larimore, "Theory and Design of Adaptive Filters", New York: Wiley, 1987.

[10] The Network Simulator – ns-2: http://www.isi.edu/nsnam/ns/index.html

[11] A. Sideri, E. Varvarigos, "New Assembly Techniques for Optical Burst Switched Networks Based on Traffic Prediction", ONDM 2007.

[12] D. Morato, J. Aracil, L. A. Diez, M. Izal, E. Magana, "On linear prediction of Internet traffic for packet and burst switching networks", ICCCN, 2001.

[13] S. Ostring, H. Sirisena, "The influence of long-range dependence on traffic prediction", International Conference on Communications, 2001.

[14] Optical burst switching simulator based on ns-2: http://wine.icu.ac.kr/~obsns/index.php