

Supporting QoS with Look-ahead Window Contention Resolution in Optical Burst Switched Networks

Farid Farahmand

Department of Electrical Engineering
The University of Texas at Dallas
Richardson, Texas, TX 75083-0688
e-mail: ffarid@utdallas.edu

Jason P. Jue

Department of Computer Science
The University of Texas at Dallas
Richardson, Texas, TX 75083-0688
e-mail: jjue@utdallas.edu

Abstract— Optical burst switching (OBS) has been proposed as a competitive hybrid switching technology to support the next-generation optical Internet. This paper addresses the problem of contention in OBS networks and introduces a new contention resolution algorithm called Look-ahead window Contention Resolution (LCR) that can also support service differentiation. Simulation results show that the performance of LCR is competitive to existing contention resolution mechanisms in terms of reducing burst loss.

Keywords- Contention Resolution; Look-ahead Window; Optical Burst Switching; Quality of Service (QoS)

I. INTRODUCTION

Exponential growth in Internet traffic has led to IP-over-WDM as the core architecture for the next-generation optical Internet. Optical burst switching (OBS) has been proposed as an efficient way to satisfy the future bandwidth requirements of such networks [1]. In OBS, IP packets are assembled into super-size packets called data bursts. These bursts are transmitted following a burst header packet (BHP) after some offset time [2]. Each BHP contains routing, scheduling, and packet priority information and is processed electronically prior to its data burst arrival. Consequently, when the data burst arrives, it can “cut-through” the switch on the pre-assigned path with minimum processing. Different signaling and scheduling mechanisms for reserving and releasing resources have been proposed for OBS.

First-Fit, Horizon [3], Latest Available Unscheduled Channel (LAUC), and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) [4], are among the proposed scheduling algorithms. In both LAUC and LAUC-VF scheduling algorithms, a burst chooses the unused channel that becomes available at the latest time. When void filling (VF) is allowed, gaps between two scheduled data bursts can also be utilized. In these schemes the data burst reservation time starts at the beginning of the actual burst arrival and lasts until the end of the burst.

A major concern in OBS networks is high contention and burst loss. Typically, there are two main sources of burst loss: contention on the outgoing data channels and contention on the outgoing control channel. In this article we focus on output data channel contention, which occurs when the total number of data bursts going to the same output port at a given time is larger than the available channels on that port. Contention is aggravated

when the traffic becomes bursty and when the data burst duration varies and becomes longer.

Contention and loss may be reduced by implementing contention resolution policies. There are different types of contention resolution techniques, such as time deflection (using buffering) [5], space deflection (using deflection routing) [6], and wavelength conversion (using wavelength converters) [7]. When a contention cannot be resolved by any one of these techniques, one or more bursts must be dropped. The policy for selecting which bursts to drop is referred to as the soft contention resolution policy.

A soft contention resolution algorithm may be utilized in conjunction with a scheduling algorithm to protect high priority bursts while reducing the overall burst loss rate. Thus, the contention resolution algorithm is invoked only when no available unscheduled channel can be found for a BHP request.

Two well-defined soft contention resolution algorithms have been proposed. One is based on dropping the latest arrival and the other is based on dropping only the portions of the burst involved in contention. In this paper we elaborate on the performance of each of the above schemes as well as their QoS supporting capacity. We also introduce two new algorithms capable of handling service differentiation: look-ahead and shortest drop contention resolution. The main contribution of this paper is to provide an efficient algorithm in order to resolve contention while minimizing burst loss.

In the subsequent sections we first briefly describe the system configuration under consideration. Then, we provide detailed descriptions pertaining to several proposed contention resolution algorithms. Finally, we present performance results for each of the introduced algorithms.

II. DESCRIPTION OF SOFT CONTENTION RESOLUTION ALGORITHMS

In this section we describe the network under study and details of different soft contention resolution algorithms.

A. Network assumptions

The network under discussion in this paper consists of a number of edge nodes connected to a core optical network with no buffering capacity. We assume that each link has a single

control channel and multiple data channels. A detailed architecture of the core and edge nodes is provided in [4] and [8]. The data burst transmission scheme can be either slotted or unslotted. In this paper, we assume slotted transmission in which data bursts and their corresponding BHPs are transmitted only on the slot boundaries. Consequently, the offset time and the duration of a data burst will be interpreted in units of slots. Furthermore, we assume that incoming data bursts have different service types with different QoS requirements. The maximum burst length is assumed to be the same for all data bursts.

B. Latest Arrival Drop Policy (LDP)

The simplest soft contention resolution policy is the latest-arrival drop policy (LDP). In LDP, the algorithm searches for an available unscheduled channel (as in LAUC-VF), and if no such channel is found, the latest incoming data burst will be discarded. Although the processing speed of BHPs in the LDP scheme is attractive, the main disadvantage of this technique is that it has relatively poor performance with respect to data loss when no buffers are utilized.

Inherently, LDP is not capable of differentiating packets with different priority types. A novel scheme proposed by [9] suggests that giving extra offset time to high priority data bursts can ensure their early reservations. This approach is known as offset-time-based QoS. The extra offset time must be large enough to ensure that the blocking of high-priority bursts by any lower-priority burst is minimized. Therefore, offset-time-based QoS is a tradeoff between guaranteeing lower loss for high priority data bursts and increasing their end-to-end delay.

C. Look-ahead Window Contention Resolution (LCR)

The look-ahead contention resolution algorithm takes advantage of the separation between the data bursts and the burst header packets. By receiving BHPs one offset time (Δ) prior to their corresponding data bursts, it is possible to construct a look-ahead window (LaW) with a size of W time units. Such a collective view of multiple BHPs results in more efficient decisions with regard to which incoming bursts should be discarded or reserved. On the other hand, at each hop, the BHPs must be stored for duration of W time units before they are retransmitted (thus requiring $\Delta \geq W$). Fiber delay lines (FDLs) can be used on each hop to delay data bursts by W time units to maintain the original offset time.

Figure 1(a) shows an example of the received BHPs for data bursts that are destined for the same switch output port with two available channels. Without loss of generality, we assume slotted transmission with window size $W=2 \times L_{max}$ time slots ($t19$ through $t27$), where L_{max} is the maximum data burst duration in units of time slots. Using the received burst header information, a burst window can be constructed (Figure 1(b)) to describe the state of the switch one offset time later ($t19 + \Delta$ through $t27 + \Delta$). Once the burst arrival times within the burst window are determined, the LCR algorithm is applied to the entire burst window range. We define TS_o and TS_w as the starting and ending slots of each burst window, respectively. The algorithm finds the contending slots and then identifies which bursts should be discarded and which should be scheduled. Note that

hard decisions are only applied to bursts when they reach the front of the burst window at TS_o (e.g., burst B_1 in Figure 1(b)). The LaW and burst window are advanced one slot at a time.

The look-ahead contention resolution algorithm can be divided into three basic steps: (a) collecting all BHPs destined to the same output port and creating a look-ahead window of size W ; (b) determining the contention regions (slots), CR , in each corresponding burst window; (c) applying a heuristic algorithm to decide which of the contending data bursts within the burst window must be discarded.

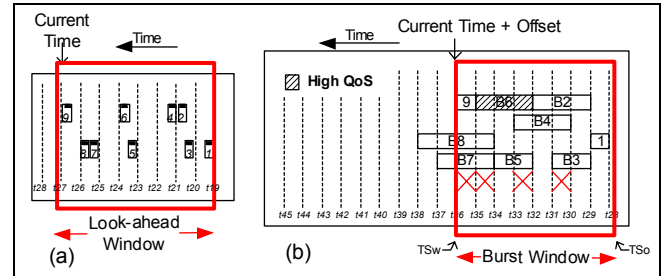


Figure 1. Look-ahead and burst windows for all bursts going to the same switch output port with 2 channels; $\Delta=9$, $W=8$, $L_{max}=4$; \times indicates contending regions.

Once the LaW is constructed and the arrival and departure times of the incoming bursts are determined, the contention resolution problem can be reduced to the following: if the number of bursts directed to the same outgoing port on the switch exceeds its available channels, how can we resolve the contention while minimizing the BLR?

The contention problem can be solved by creating an auxiliary directed graph $\mathbf{G}=(\mathbf{V}, \mathbf{E})$ representing all the bursts in the LaW. In this representation the finite set \mathbf{V} of vertices (nodes) identifies the starting and ending times of the bursts. That is $\mathbf{V} = \{t_s(1), t_s(2), \dots, t_s(wb), t_e(1), t_e(2), \dots, t_e(wb)\}$, where $t_s(i)$ and $t_e(i)$ are the starting and ending times of data burst i , respectively, and wb is the number of bursts in the LaW. The finite set \mathbf{E} of directed edges includes one directed edge for each burst. An edge exists between vertices $t_s(i)$ and $t_e(i)$ to represent burst i . The weight of each edge is equivalent to the duration of its corresponding data burst, $L(B_i)$. Furthermore, we define a set of contention regions, $\mathbf{CR} = \{CR_1, CR_2, \dots, CR_u\}$ within the burst window, where CR_i extends from $t_s(m)$ to $t_e(n)$. By finding the least-cost path from the beginning of the first contention region to the end of the last contention region (CR_1 through CR_u), we can find a set of data bursts that, if dropped, can resolve the contention while minimizing data loss.

In order, to implement the shortest-path algorithm we need to alter the original digraph \mathbf{G} such that it is *connected*. Therefore, we introduce a set of zero-weight directed edges ($Z_{k+1,k}$) between adjacent nodes $k+1$ and k when the following conditions exist: (a) The adjacent nodes between contention resolution regions are not strongly connected. This can occur when, for example, there is a short halt in data burst transmissions. (b) The outer nodes of contention regions are disconnected with their non-contending adjacent nodes. This is due to the fact that only parts of data bursts may be involved in contention and thus the shortest-path solution may stretch beyond the contending slots.

The adjacent nodes within contention regions may also be disconnected. This is because overlapping data bursts can end on different time slots. Thus, in order to ensure graph connectivity within contention regions we can add directed return paths, $N_{k+1,k}$, between adjacent nodes $k+1$ and k . We now describe the scheme in which the weight of return paths can be determined. Let us define the contention degree, d_{TS} , as the number of unsuccessful bursts contending for an outgoing port in time slot TS. The shortest-path solution must remove as many as d_{TS} overlapping data bursts on each time slot TS in the window. This can be emphasized by having i negative-weight directed paths between adjacent nodes $k+1$ and k ($N_{k+1,k}^{(i)}$, with $i=0, 1, \dots, d_{TS}-1$). Having $i=0$ implies that only one contending data burst must be removed. Thus, in this case, we can assume that the weight of the return path is zero ($|N_{k+1,k}^{(0)}| = |Z_{k+1,k}| = 0$).

On the other hand, when $i \geq 1$ the weight of each return path can be defined as

$$|N_{k+1,k}^{(i)}| = (-1) \times |E_{k,m}^{(j-i-1)}|, \quad (1)$$

where $|E_{k,m}^{(j)}|$ is the weight of the outgoing edge, $E^{(j)}$, from node k to another arbitrary node m such that

$$|E_k^{(j+1)}| > |E_k^{(j)}|.$$

The resulting connected digraph, including the zero and return directed paths, can be represented by $\mathbf{G}^* = (\mathbf{V}, \mathbf{E}^*)$. The shortest-path algorithm can now be solved for \mathbf{G}^* . Consequently, the solution can be obtained by implementing the Bellman-Ford algorithm which has a complexity of $\mathcal{O}(|\mathbf{V}| \cdot |\mathbf{E}^*|)$. Other variants of the Bellman-Ford algorithm can also be considered. In either case, special care must be taken to ensure that no negative cycles occur. Note that if the obtained shortest-path solution does not eliminate all the contending bursts, the algorithm must be repeated for the remaining contention regions.

Once the shortest-path algorithm is completed, a set of data bursts, $\mathbf{P} = \{B_x, B_y, \dots\}$, is obtained for possible discard. Only the bursts with starting time equal to TS_0 can be permanently dropped.

The LCR algorithm can be readily modified to support service differentiation. Let us assume that the class type for a data burst B_i^c is defined by c , with c_{max} being the lowest priority level. The starting and ending slots of burst B_i are denoted by $t_s(i)$ and $t_e(i)$, respectively. In this case, the weight of the edge $E_{t_s(i), t_e(i)}$ connecting node pairs $t_s(i)$ and $t_e(i)$ in graph $\mathbf{G}^* = (\mathbf{V}, \mathbf{E}^*)$ can be a function of the duration and the priority level of burst i . That is

$$E_{t_s(i), t_e(i)} = [c_{max} - c] \cdot L_{max} + L(B_i^c). \quad (2)$$

Note that when all bursts have the same class priority, the edge weights become equivalent to burst durations.

Several advantages can be attributed to the QoS-enabled LCR. For instance, it can support unlimited number of classes of service without requiring extra offset time. The LCR mechanism can offer absolute as well as proportional differentiation. In

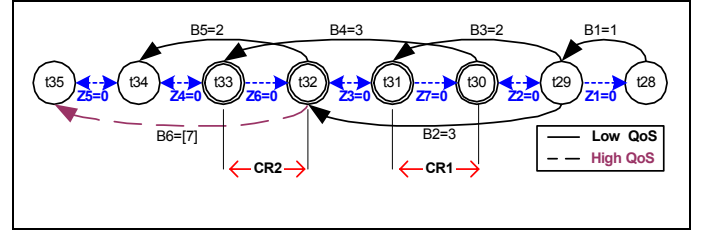


Figure 2. Directed graph, $\mathbf{G}^* = (\mathbf{V}, \mathbf{E}^*)$, partial representation of example shown in Figure 1; for simplicity bursts with $t_s(i)$ beyond $t35$ are not shown; $L_{max}=4$; B_6 is assumed to have high priority ($c=1$) and $c_{max}=2$.

absolute differentiation the possibility of a high-priority burst being blocked by any lower priority burst is eliminated. On the other hand, in proportional differentiation the dropping criteria will be based on the relative length and priority level of data bursts. In such a scheme, it is possible that between a short duration high priority burst and a long duration low priority burst, the one with higher priority will be discarded. Clearly, in terms of complexity, minimal additional steps are required to enable service differentiation in LCR.

At this point, we demonstrate the LCR approach by referring to the example shown in Figure 1. We start by creating a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The set of bursts within the look-ahead window is represented by $\mathbf{B} = \{B_1, B_2, \dots, B_{wb}\}$, with $wb = 9$. Thus, there will be 9 edges with 11 distinct nodes in \mathbf{G} , where $\mathbf{V} = \{t28, t29, t31, \dots, t38\}$ and $\mathbf{E} = \{(t28, t29), (t29, t31), (t29, t32), \dots\}$. Moreover, $\mathbf{CR} = \{CR_1, CR_2\}$, where $CR_1 = (t30, t31)$ and $CR_2 = (t32, t33)$. Each edge $(t_s(i), t_e(i))$ is assigned a weight representing the burst i duration and its priority level. Assuming $c_{max} = 2$, the weight of the edge $(t32, 35)$ representing B_6 will be 7.

Figure 2 depicts the modified digraph \mathbf{G}^* after adding the zero and return paths. Note that directed return paths of $N_{t33, t32}^{(0)}$ and $N_{t31, t30}^{(0)}$ are equivalent to zero-weight paths of Z_6 and Z_7 , respectively. This is because the degree of contention in these time slots is one. Also, note that $Z_{t28, t29}$ has been replaced by the edge representing B_1 . Solving the shortest-path problem, we find $\mathbf{D} = \{B_4, B_7\}$. This indicates that by discarding B_4 and B_7 all contentions can be eliminated. However, since none of these data bursts arrive at the starting slot of the window ($TS_0=t28$) no burst will actually be dropped until the window reaches the start of either burst.

D. Shortest Burst Drop Policy (SDP)

In order to reduce the end-to-end data burst delay in LCR, different variations of this algorithm can be considered. The tradeoff, of course, will be performance degradation. For example, if we reduce the window size to L_{max} , the incoming data bursts can experience shorter per hop delay, but the advance viewing capacity of the window will be decreased. The window size can also be minimized to a single slot. In this scheme, each incoming burst slot will be checked, and upon detecting contention, the lower priority burst with the shortest duration and latest arrival time will preferentially be dropped. This allows BHPs to be processed and transmitted soon after they are received. We call this scheme the shortest drop policy

(SDP). One drawback of such a policy is its potential over-reserving of resources, since some earlier reservations may be eliminated later. In other words, the reservation for BHP_i can potentially be cancelled within the next Δ_i slots before the associated data burst arrives.

In terms of supporting class differentiation, SDP can support unlimited number of priority levels and requires no extra offset assignments for bursts with higher service requirements. It also guarantees complete class isolation. In addition, SDP offers proportional differentiation, as described above.

E. Segmentation Drop Policy (SEG)

The basic assumption in this scheme is that each transmitted data burst consists of individual independent segments such as slots. Therefore, if contention occurs, only the segments of the lower priority burst involved in the contention will be removed. Details of this mechanism, known as Segmentation, along with its variations are described in [10]. Although the QoS-enabled Segmentation algorithm appears to be straightforward, the hardware implementation in terms of burst assembly and disassembly, as well as overhead insertion and extraction, can be complex.

III. PERFORMANCE COMPARISON

In this section we present the simulation results for each of the introduced policies. We start by applying these algorithms to a simple example where packets can have low or high priority levels.

A. Numerical comparison between different soft contention resolution algorithms

Figure 3 shows the expected incoming bursts from different ingress ports between time slots 19 through 33. We assume all bursts (B_1 - B_8) have the same destination address and each outgoing port of the switch has only 2 wavelengths ($N=2$). First, we implement the LAUC-VF scheduling algorithm without utilizing any contention resolution algorithm. We assume high-priority data bursts, namely B_4 and B_6 , were reserved much in advance. In this case the latest contending request will be dropped. Thus, discarding B_1 , B_7 , and B_8 can be considered as the worst-case outcome.

Using Segmentation, individual slots of B_2 , B_3 , B_5 , and B_8 can be isolated and dropped. In this case, we ignore the impacts of extra overhead requirements in order to divide data bursts into independent segments.

We then consider the LCR algorithm with $W=2 \times L_{max}=14$. Therefore, the weights associated to the high-priority data burst B_4 and B_6 will be 12 and 9, respectively. The edge weights representing low priority bursts will be equivalent to their durations. Table 1 summarizes the resulting performance using different contention resolution algorithms. Note that LPD can potentially result in the worst-case performance while the SEG technique provides an upper bound on performance.

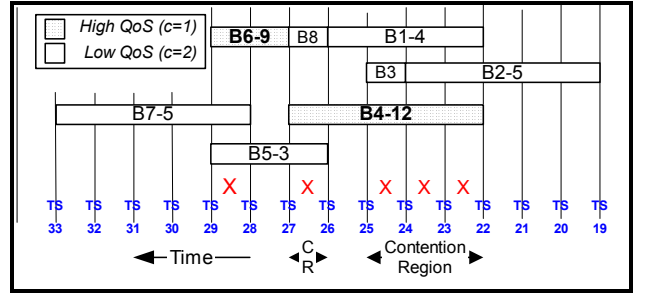


Figure 3. Example of incoming bursts (B_i); all bursts are directed to the same destination port, $N=2$, $L_{max}=7$, $c_{max}=2$.

TABLE 1. PERFORMANCE RESULTS OF IMPLEMENTING VARIOUS CONTENTION RESOLUTIONS FOR THE EXAMPLE IN FIGURE 3

| Drop Policy | Bursts Dropped | Slots Dropped |
|-------------|----------------------------|---------------|
| LDP | B_1, B_7, B_8 | 10 |
| SDP | B_1, B_5, B_8 | 8 |
| LCR | B_1, B_5 | 7 |
| SEG | $B_2(2), B_3, B_5(1), B_8$ | 5 |

B. Simulation Results

In this section, simulation results are presented for each of the contention resolution schemes, namely the Latest Drop Policy (LDP), Shortest Drop Policy (SDP), Look-ahead Contention Resolution (LCR), and Segmentation (SEG). These results are obtained with the following assumptions:

- The network consists of a single bufferless core switch with 4 input/output ports and each port consists of 8 data channels and a single control channel.
- We assume synchronous slotted switching mechanism with the slot size granularity of 16000 bytes.
- The maximum data burst duration is 20 slots.
- Wavelength conversions are utilized on all output ports of the switch.
- Data bursts can have three distinct priority levels, $c=1, 2, 3$, with distribution ratio of 10%, 30%, and 60%, respectively.
- Inter-arrival times between BHPs are exponentially distributed.
- Source-destination pairs are assigned based on a uniform distribution.
- Offsets between BHPs and their associated data bursts are fixed.

We represent the simulation results in terms of utilization, and burst loss rate. Utilization, G , is the ratio of the total number of burst slots generated in the network per unit time. The burst loss rate, BLR, is the percentage of burst slots that are sent by the source but never received by the destination.

Figure 4(a) shows the overall performance of LCR compared to SEG and LDP algorithms. Note that, as expected, SEG and LDP provide the upper and lower bounds on performance, respectively. On average the LCR performs about 20% better than LDP. The impact of the window size in LCR is shown in Figure 4(b). Note that as the window size increases, the burst loss rate decreases. This figure suggests that the bulk of improvement is accomplished when W is between L_{max} and $2L_{max}$.

Further simulation results (not shown here) indicate that the LCR performance is slightly better than SDP due to its deeper viewing ability. However, in a multi-switch system the overall burst loss rate using the SDP algorithm can actually be reduced due to its potential over-reservation. As we noted before, a major issue with LCR is its per hop delay, which is equivalent to W time slots. The SDP can be considered as a reasonable tradeoff between reducing delay and slightly lowering the performance. The resulting burst loss ratio for all three classes of service using LCR is shown in Figure 5(a). These results can be compared with Figure 5(b), which depicts loss for individual classes using LDP. The performance results of the lower two classes are transposed on each other in Figure 6.

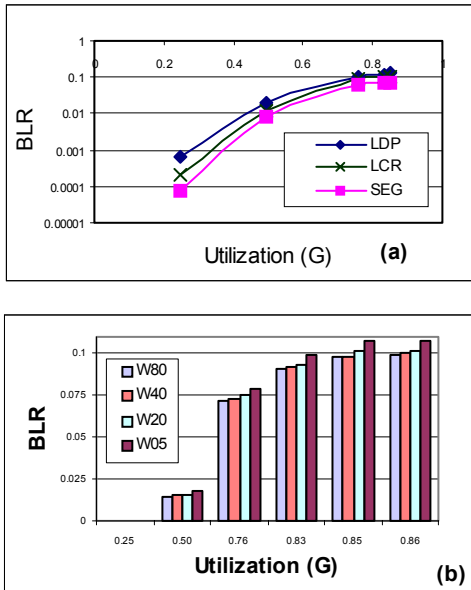


Figure 4. (a) Overall BLR performance using different contention resolution schemes with $W=40$, $L_{max}=20$ slots; (b) LCR overall performance with window sizes (W): 5, 20, 40, 80 with $L_{max}=20$.

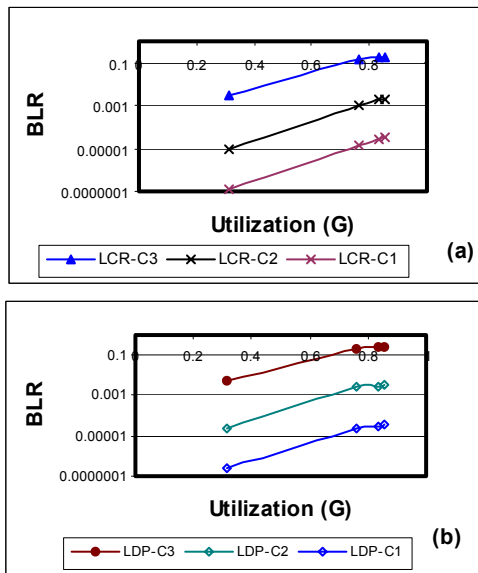


Figure 5. BLR performance for all three classes using (a) LCR (b) LDP. C_i indicates the highest priority level.

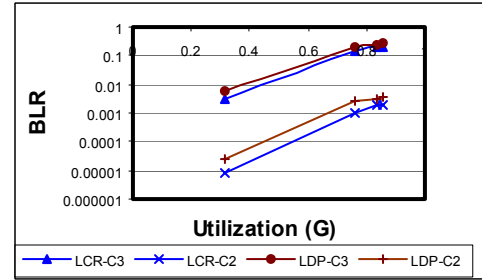


Figure 6. BLR comparison of classes 2 and 3 in LDP and LCR.

IV. CONCLUSION

We presented several contention resolution algorithms for optical burst switching networks, namely the Latest Drop Policy, Shortest Drop Policy, Look-ahead Contention Resolution, and Segmentation.

We discussed each algorithm and its implementation complexity and examined its performance in terms of burst loss rate for different classes of service. Simulation results show that the look-ahead contention resolution algorithm can readily support service differentiation and offers high overall performance with moderate complexity. The LCR algorithm can be modified to reduce the total end-to-end burst delay at the cost of slightly lowering the performance.

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, no.1, pp.69-84, Jan. 1999.
- [2] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks," *IEEE/LEOS Conf. on Technologies for a Global Information Infrastructure*, pp. 26-27, Aug. 1997.
- [3] J.S. Turner, "Terabit burst switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3-16, Jan. 1999.
- [4] Y. Xiong, M. Vanderhoute, and H.C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838-1851, Oct. 2000.
- [5] C. Gauger, "Dimensioning of FDL Buffers for Optical Burst Switching Nodes," *Proceedings, Optical Network Design and Modeling (ONDM) 2002*, Torino, Italy, 2002.
- [6] X. Wang, H. Morikawa, and T. Aoyama, "A Deflection Routing Protocol for Optical Bursts in WDM Networks," *Proceedings, Fifth Optoelectronics and Communications Conference (OECC) 2000*, Makuhari, Japan, pp. 94-95, July 2000.
- [7] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "All-optical packet switching for Metropolitan Area Networks: Opportunities and Challenges," *IEEE Communication Magazine*, vol. 39, no. 3, pp. 142-148, March 2001.
- [8] F. Callegati, H.C. Cankaya, Yijun Xiong, and M. Vandenhouete, "Design issues of optical IP routers for Internet backbone applications," *IEEE Communications Magazine*, vol. 37, no 12, pp. 124-128, Dec. 1999.
- [9] M. Yoo and C. Qiao, "Supporting Multiple Classes of Services in IP over WDM Networks," *Proceedings, IEEE GLOBECOM 1999*, Brazil, pp. 1023-1027, Dec. 1999.
- [10] V. Vokkarane, J. Jue, and S. Sitaraman, "Burst Segmentation: An Approach for Reducing Packet Loss In Optical Burst Switched," *Proceedings, IEEE International Conference on Communications (ICC) 2002*, New York, NY, vol. 5, pp. 2673-2677, April 2002.

