Look-ahead Window Contention Resolution in Optical Burst Switched Networks

FARID FARAHMAND and JASON P. JUE , *Member, IEEE*

Abstract-- **In this article we introduce a new contention resolution algorithm called Look-ahead Window Contention Resolution (LCR). This unique contention resolution approach can be used in conjunction with a scheduling technique to reduce the packet loss in an optical burst switched network. We analyze the performance of LCR by means of simulations and compare it with other major existing contention resolution policies.**

Index Terms—**Burst Loss Rate, Contention Resolution Algorithm, Look-ahead Window, Optical Burst Switching.**

I. INTRODUCTION

Over the past decade the exponential growth in Internet traffic and data communications has fueled an increasing interest in optical packet switching [1]. In addition to ultimately enabling a transmission capacity of hundreds of Gb/s per fiber, all-optical packet-switched WDM networks are transparent to any bit rate and readily scalable as the demand for bandwidth increases. However, problems with synchronization and lack of optical memory (flip-flops) and buffering have made optical packet switching deployment impractical for field implementation.

Optical burst switching (OBS) has been proposed as an alternative paradigm for tera-bit networks for overcoming such technological barriers [2]. The basic idea in OBS is to assemble IP packets into super-size packets called data bursts (DB) and to transmit these bursts following a burst header packet (BHP) after some offset time [3]. Each BHP contains routing and scheduling information and is processed electronically prior to its data burst arrival. As a result, the switch is set up in advance; thus, when the data burst arrives it can "cut-through" the switch with minimum processing. Different signaling and scheduling mechanisms describing the manner in which connections are established and resources are reserved and released have been proposed for OBS.

First-Fit, Horizon [4], Latest Available Unscheduled Channel (LAUC), and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) [5], 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 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) [6], space deflection (using deflection routing) [7]-[8], wavelength conversion (using wavelength converters), and soft contention resolution (using different contention resolution algorithms). Clearly, a combination of such techniques can be very effective.

Using buffering in the core switches may not be viable, since the hardware complexity and high cost of such devices make them less attractive and limits their practicality. Space deflection can result in inefficient routing and potentially a high number of collisions. Furthermore, it results in high end-to-end delay and possible packet reordering, neither of which may be acceptable for many applications. Wavelength conversion on output ports is a very efficient approach for resolving contention and adds an additional dimension (in addition to time and space) to contention resolution.

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 reduce the overall burst loss rate, BLR, and consequently, enhancing link utilization. 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 and studied in earlier literature. One is based on dropping the latest arrival [5] and the other is based on dropping only the portions of the burst involved in contention [10] (known as Segmentation). In this paper we elaborate on the performance of each of the above schemes and introduce two new algorithms, namely Look-ahead and Shortest-drop contention resolution. The main contribution of this paper is to

---

F. Farahmand is with the Department of Electrical Engineering, The University of Texas at Dallas, Richardson, Texas, U.S.A (e-mail: ffarid@utdallas.edu).

J. P. Jue is with the Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, U.S.A (e-mail: jjue@utdallas.edu).

provide an efficient contention resolution policy in order to resolve contention while minimizing the overall BLR.

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 CONTENTION RESOLUTION ALGORITHMS

In this section we describe the details of several contention resolution techniques, which can be implemented in optical burst switching. In the following discussions we ignore service differentiation and do not consider the algorithm's fairness. We assume that all dropping policies contain a round-robin mechanism in which discarded bursts are equally distributed among all channels.

A. Network assumptions

In the following subsections, without loss of generality, we assume a single core switch system with no buffering capacity and P edge nodes connected to the switch. Furthermore, we assume that each link has a single control channel, and N data channels. A detailed architecture of the core switch nodes and edge nodes is provided in [5] and [9]. The data burst transmission scheme can be either slotted or unslotted. However, for providing a better description of algorithms we assume slotted transmission in which data bursts and their BHPs are transmitted only on the slot boundaries. Consequently, offset time (Δ) and data burst duration (LBi) will be interpreted in terms of data burst slots.

B. Latest Arrival Drop Policy (LDP)

The simplest algorithm to resolve contention is based on dropping the most recently arrived burst when there is no unscheduled channel available. 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. One major advantage of LDP is that is has fast processing speeds and low end-to-end burst delay. The scheduling decisions are completed as soon as BHPs are processed. The main disadvantage of this technique is that it has relatively low performance when no buffers are utilized.

C. Look-ahead Window Contention Resolution

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. Having 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 a duration of W time units before they are retransmitted (thus requiring Δ ≥ W). Clearly, one way to maintain the original offset time is to delay data bursts by W time units by using fiber delay lines (FDLs) on each hop.

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×Lmax time slots (t19 through t27) where Lmax is the maximum data burst duration in units of time slots. Using the received burst header information, Figure 1-b can be constructed to describe the state of the switch one offset time later (t19 + Δ through t27 + Δ). Once the burst arrival times within the burst window are determined, the contending slots can be found. Using the LCR algorithm it is possible to identify which bursts should be discarded and which should be scheduled. However, the data bursts are actually dropped or scheduled only when the starting time of a burst is equal to the start of the burst window (in this case B1). After the LCR process is completed for the look-ahead window, the starting time of the window, TSw, is advanced to the next slot and may include new BHPs (t20 through t28). Scheduled requests are irreversible and cannot be changed by the future requests.

Two schemes can be considered to define the range in which the contention resolution algorithm can be applied within the window. First, we can consider the entire window range (TSo-TSw) and include all bursts with starting time less than TSw. Definite decisions can only be applied to the bursts with starting time equal to TSo. The LAW is advanced one slot at a time. In the second scheme, we define a region called the Resolution Region (RR). The RR is bounded by TSo and TSr indicating its starting and ending points in time, respectively. Clearly, TSo is always the same as the starting point of the LAW. We define TSr as the starting point of the earliest burst whose duration extends beyond the end of the window. Once the algorithm is performed for the bursts within the RR, the window is advanced as many as TSr slots. For example, in Figure 1-b the RR ranges from t28 to t34. Thus, the next window will start from TSr = t34. It can easily be seen that, although the end-to-end delay in the first approach is slightly larger than the second approach. In fact, the first approach results in higher overall performance. Therefore, in the remainder of this document, we only consider the first approach.

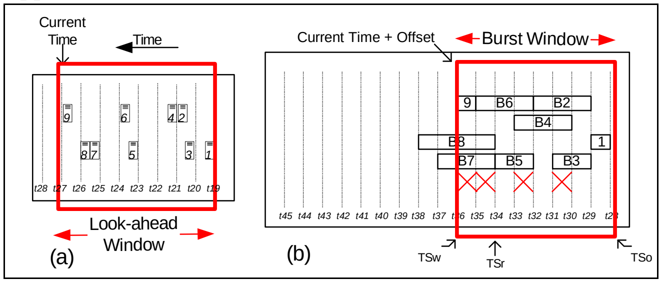


Figure 1. Constructing the look-ahead window for all bursts going to the same switch port; Δ=9, W=8, P=2, N=2; X indicates contending regions.

Description of the algorithm: 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

TABLE 1. DESCRIPTION OF THE LCR ALGORITHM - STEPS A AND B

.

a) *Initialization:*
    1- *Assume W = 2x Max_Burst_length= 2xLmax*
    2- *Increment time slots until a full size (W-time slots) look-ahead window is reached*
    3- *Map the Burst_Window = [TSo - TSw]; where TSw = Current_Time + Offset*
    4- *Let **B** = { Bi(L,ts) }; where Bi is a burst within the burst window with duration L and starting time ts*

b) *Within the look-ahead burst window define the resolution region: RR = [TSo – TSw]*

c) *Identify all contention regions within the burst window: **CR** = {CR1, CR2, ...CRu}, where u ≤ W/2*

d) *Resolve contention regions within W by identifying which bursts can be discarded. Only the bursts with ts = TSo can be marked for drop. (this is STEP C - details are described in Table 2)*

e) *Schedule un-marked bursts with starting time, ts = TSo*

f) *Advance the window to the next time slot: TSo = TSo + 1*

contention regions (slots) in each corresponding window; (c) applying a heuristic algorithm to decide which of the contending bursts within the window must be discarded. Table 1 describes the details of steps (a) and (b).

Once the look-ahead window (LAW) is constructed and the arrival times of the incoming bursts along with their durations are determined, the contention resolution problem can be reduced to the following: if there are more than N bursts directed to the same outgoing port on the switch, how can we (near) optimally resolve the contention while minimizing the BLR? The contention resolution problem can be solved by creating an auxiliary directed graph and applying a shortest-path algorithm to the graph. We describe the details of this approach in the following paragraphs.

The content of the LAW can be represented by a digraph ***G*** = (N, A). The parameter N is defined as the set {($t_s(1)$, $t_s(2)$, … $t_s(q)$, $t_e(1)$, $t_e(2)$, …, $t_e(q)$}, where $t_s(i)$ and $t_e(i)$ are the starting and ending times of burst B_i, respectively and q is the number of bursts in the LAW. Furthermore, the set of edges A represents a collection of ordered pairs of distinct nodes from N with a weight equivalent to the duration of burst B_i, LB_i. Given a set of contention regions, ***CR*** = {CR_1, CR_2, ...CR_u} within the LAW, where CR_i extends from $t_s(m)$ to $t_e(n)$, we can implement any standard centralized shortest-path algorithms to find the shortest bursts going through CR_1 to CR_u such that there are no more port contentions. Thus, the shortest-path problem is simply to find a set of shortest bursts, ***D***, with the directed path going through ***CR***.

In order to solve the shortest-path problem, we need to alter the original digraph ***G*** such that it is *connected*. Therefore, we developed a series of simple rules to interconnect the adjacent nodes (K-1, K, K+1) together. These rules are described in part b of Table 2. The zero-path directed connections ($Z_{k,k+1}$ or $Z_{k+1,k}$) are zero weighted arcs between adjacent nodes and are required for graph connectivity purpose, since in many cases, the contention regions may be disjoint from each other. Also, within a contention region, adjacent nodes may not be connected. Negative-path connections ($N_{k+1,k}$) are used to

TABLE 2. FINDING THE SHORTEST BURSTS IN THE LAW TO RESOLVE CONTENTION – STEP C

.

a) *Construct a directed graph (**G**) using the burst window:*
    1- *Assign a node pair (ts(i), te(i)) such that ts(i) and te(i) are the starting and ending time of each burst, Bi, respectively*
    2- *Represent the distance between each node pair by the burst duration, LBi*

b) *Interconnect adjacent nodes:*
    1- *Add directed zero-path connections, $Z_{k,k+1}$ and $Z_{k+1,k}$, between all adjacent node pairs [(k, k+1) and (k+1, k)] within the look-ahead burst window*
    2- *Remove $Z_{k,k+1}$ and $Z_{k+1,k}$ if node k is the starting node of a contention region*
    3- *Remove $Z_{k,k-1}$ and $Z_{k-1,k}$ if node k is the ending node of a contention region*
    4- *Remove $Z_{k,k+1}$ if node k is within a contention region*
    5- *Assign a negative-path, $N_{k+1,k}$, from k+1 to k, if node k is within a the contention region*
    6- *Replace $Z_{k,k+1}$ with any non-zero-directed path from k to k+1*

c) *Generate the cost matrix Cij for the new directed graph **G`***

d) *Using a shortest path algorithm find the shortest bursts, from CR1 to CRu, **P** ={Bi, Bj, …}*

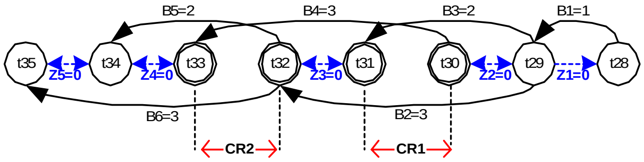
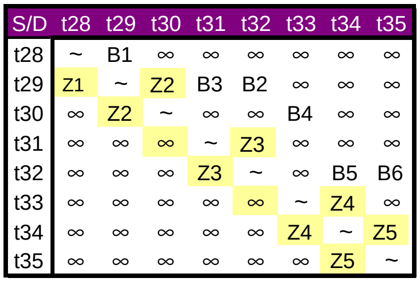


Figure 2. Directed graph representation of the example shown in Figure 1; note that for simplicity, bursts with ts(i) beyond t35 are not shown since they have no impact on B1.



S/D	t28	t29	t30	t31	t32	t33	t34	t35
t28	~	B1	∞	∞	∞	∞	∞	∞
t29	Z1	~	Z2	B3	B2	∞	∞	∞
t30	∞	Z2	~	∞	∞	B4	∞	∞
t31	∞	∞	∞	~	Z3	∞	∞	∞
t32	∞	∞	∞	Z3	~	∞	B5	B6
t33	∞	∞	∞	∞	∞	~	Z4	∞
t34	∞	∞	∞	∞	∞	Z4	~	Z5
t35	∞	∞	∞	∞	∞	∞	Z5	~

Figure 3. Distance matrix, ***C***$_{ij}$, representing the directed graph shown in Figure 2, after applying the connectivity rules in Table 2 to interconnect adjacent nodes together, ***G`*** = (N, A`).

distinguish between overlapping bursts and a single burst with the same total length. In general, it is preferred to discard several overlapping bursts within the contention regions.

Once the resulting connected digraph, ***G`*** = (N, A`), is defined, its associated distance matrix, ***C***$_{ij}$, can be constructed. If (i,j) is not an arc of the graph we denote its weight, d_{ij}, as infinity. The computation complexity to find the shortest set of bursts in each window depends on the choice of the shortest-path algorithm. Variants of the Bellman-Ford and Dijkstra algorithms appear to be practical and efficient. Note that multiple iterations of the LCR algorithm may need to be performed in the LAW in order to resolve all data burst contentions.

We demonstrate the above approach by referring to the example shown in Figure 1. We start by creating a directed graph $\boldsymbol{G} = (N, A)$. The set of bursts within the look-ahead window is represented by $\boldsymbol{B} = \{B_1, B_2, \dots B_q\}$, with q = 9. Thus, there will be 9 arcs with 11 distinct nodes in $\boldsymbol{G}$. Each arc $(t_s(i), t_e(i))$ is assigned a weight representing the burst duration, LBi. Consequently, we will have N = {t28, t29, t31, …, t38} and A = {(t28, t29), (t29, t31), (t29, t32), …}. In this case CR1 = (t30, t31), CR2 = (t32, t33), etc.

Figure 2 shows the resulting digraph, $\boldsymbol{G}$`, for the example shown in Figure 1 after applying the connectivity rules between adjacent nodes. Note that $Z_{t28,t29}$ has been replaced by B1 length. The corresponding distance matrix, $\boldsymbol{C}_{ij}$, of the digraph is shown in Figure 3. Solving for the shortest-path, as described in Table 2, we will have $\boldsymbol{D} = \{B_4, B_7\}$. This indicates that by discarding B4 and B7 all contentions can be eliminated. However, since none of these data bursts arrive at the starting slot of the window (TSo=t28) no burst will actually be dropped until the window reaches the start of either burst.

D. *Shortest Burst Drop Policy (SDP)*

A less complex version of LCR algorithm, called LCR with shortest drop, can be considered. In this case, contention regions are determined within window sizes of W=Lmax. Then, in each region, the bursts with the shortest duration and latest arrival time will preferentially be dropped. In order to reduce the end-to-end data burst delay, the LCR with shortest drop algorithm can be modified such that the window size is reduced to a single slot and the contending burst with the shortest duration in each slot will be discarded. In this case BHPs are processed and transmitted as soon as they are received. We call this scheme the Shortest Drop Policy (SDP). One drawback of such scheme is its potential over-reserving of resources, since some earlier reservations may be eliminated later. Another words, the reservation for BHPi can potentially be cancelled within the next LBi slots.

E. *Segmentation Drop Policy (SEG)*

The basic idea in this scheme is to divide each burst into individual independent segments, such as slots, and remove only the segments of the burst involved in contention. Details of the Segmentation Drop Policy and its variations are described in [10]. Although implementing Segmentation appears to be straightforward, the hardware implementation in terms of burst assembly and disassembly, as well as overhead insertion and extraction, can be very complex.

III. PERFORMANCE COMPARISON

In this section we present the simulation results for each of the introduced policies in terms of BLR. We start by applying these algorithms to a simple example.

A. *Numerical comparison between different algorithms*

Figure 4 shows the expected incoming bursts from different ingress ports between time slots 20 though 31. We assume all bursts (B3-B6) have the same destination address and each outgoing port on the switch has 2 wavelengths (N=2). First, we implement the LDP using the LAUC-VF scheduling algorithm. In this case, the latest contending request will be dropped. Thus, discarding B3 and B6 can be considered as the worst-case outcome.

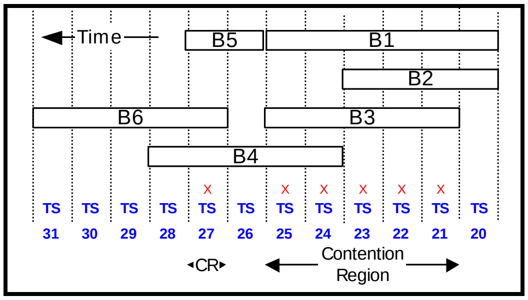


Figure 4. An example of incoming bursts (Bi); all bursts are directed to the same destination port, N=2, W=12, Lmax=6.

TABLE 3. PERFORMANCE RESULTS OF VARIOUS CONTENTION RMS ESOLUTION ALGORITHFOR THE EXAMPLE IN FIGURE 4

Drop Policy	*Bursts Dropped*	*Slots Dropped*
LDP	*B3, B6*	*10*
SDP	*B2, B4*	*9*
LCR	*B3, B5*	*7*
SEG	*B3, B6(1)*	*6*

Using Segmentation, B3 and B6 are considered for drop. However, only a single segment of B6 will actually be dropped. Note that we ignore the impacts of having extra overhead while dividing data busts into independent segments.

We then consider the LCR algorithm with W=2×Lmax=12 and the SDP algorithms. Table 3 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.

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 (P=4) and each port consists of 8 data channels (N=8) 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 input/output ports of the switch.
- All bursts have the same priority level.
- Inter-arrival times between BHPs are exponentially distributed.
- Source-destination pairs (s-d) are assigned based on a uniform distribution.

- The offset between the BHP and the data burst is 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 bursts that are sent by the source but never received by the destination.

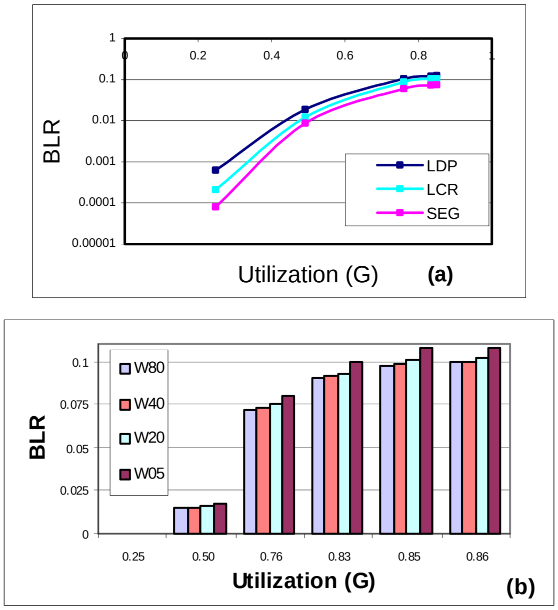


Figure 5: (a) Burst loss performance using different contention resolution schemes; Lmax= 20 slots, P = 4, N= 8, W=40 slots; (b) LCR performance under different window sizes: 5, 20, 40, 80 with Lmax= 20.

Figure 5-a shows the performance of LCR compared to the Segmentation (SEG) and Latest Drop Policy (LDP). Note that, as expected, the SEG and LDP provide the upper and lower performance limits, respectively. On average, the LCR performs about 20 percent better than LDP. The impact of the window size in BLR is presented in Figure 5-b. Note that as the window size becomes larger, the BLR improves. This figure suggests that the main BLR improvement is accomplished when W ∈ [Lmax , 2Lmax].

Figure 6 compares the performance improvement of LCR with window size W=2×Lmax slots and SDP (with W=single-slot) for various loads in terms of BLR. The LCR performance is slightly better than SDP due to LAW's deeper viewing ability. It must be noted that in a multi-switch system the overall BLR using the SDP algorithm can actually be reduced due to its potential over-reservation. However, 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.

IV. CONCLUSION

We presented several contention resolution algorithms for optical burst switching networks (OBS), namely the Latest Drop Policy (LDP), Shortest Drop Policy (SDP), Look-ahead Contention Resolution (LCR), and Segmentation (SEG). We discussed each algorithm and its implementation complexity and examined its performance in terms of burst loss rate. Simulation results show that the look-ahead contention resolution algorithm offers high performance with moderate complexity. The LCR can be modified to reduce the total end-to-end burst delay.

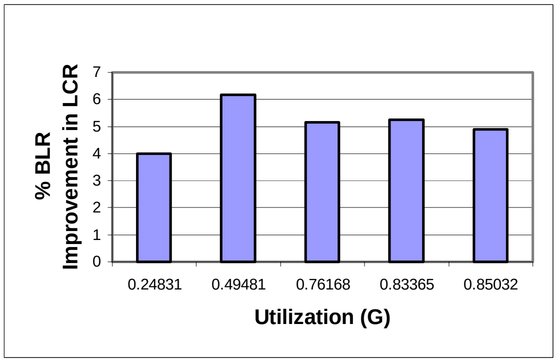


Figure 6. Comparing the performance of LCR and SDP.

REFERENCES

[1] F. Callegati, M. Casoni, C. Raffaelli, and B. Bostica, “Packet Optical Networks for High-speed TCP-IP Backbones,” *IEEE Commun. Mag.*, pp.124-129, Jan.1999.

[2] 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.

[3] 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.

[4] J.S. Turner, “Terabit burst switching,” *Journal of High Speed Networks*, Vol. 8, No. 1, January 1999, pp. 3-16.

[5] 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, October 2000, pp. 1838-1851.

[6] C. Gauger, "Dimensioning of FDL Buffers for Optical Burst Switching Nodes", Proceedings of Optical Network Design and Modeling (ONDM 2002), Torino, 2002.

[7] X. Wang, H. Morikawa, and T. Aoyama, "A Deflection Routing Protocol for Optical Bursts in WDM Networks", Proc. Fifth Optoelectronics and Communications Conference (OECC 2000), Makuhari, Japan, pp. 94-95, Jul. 2000.

[8] 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*, Feature Topic on Optical Packet Switching Networks, vol. 39, no. 3, pp. 142-148, March 2001.

[9] F. Callegati, H.C. Cankaya, Yijun Xiong, and M. Vandenhoute, "Design issues of optical IP routers for Internet backbone applications", *IEEE Communications Magazine*, vol. 37, issue 12, pg. 124-128, Dec. 1999.

[10] V. Vokkarane, J. Jue, S. Sitaraman, “Burst Segmentation: An Approach for Reducing Packet Loss In Optical Burst Switched,” Proc. ICC 2002.