Channel Scheduling Algorithms using Burst Segmentation and FDLs for Optical Burst-Switched Networks

Vinod M. Vokkarane, Guru P. V. Thodime, Venkata U. B. Challagulla, and Jason P. Jue
Department of Computer Science, The University of Texas at Dallas
Richardson, TX 75083
{vinod, vgt015000, uday, and jjue}@utdallas.edu

***Abstract*– Optical burst switching is a promising solution for terabit transmission of IP data bursts over WDM networks. One of the key components in the design of optical burst-switched nodes is the development of channel scheduling algorithms that can efficiently handle data burst contentions. Currently, traditional scheduling techniques use wavelength conversion and buffering to resolve burst contention. In this paper, we reduce packet losses by proposing a number of data channel scheduling algorithms that use burst segmentation and fiber delay lines (FDLs). The proposed scheduling algorithms are classified based on the placement of the FDL buffers in the optical burst-switched node and are referred to as delay-first or segment-first schemes. Simulation results show that these algorithms can effectively reduce the packet loss probability compared to existing scheduling techniques.**

I. INTRODUCTION

There has been a steady growth of Internet traffic during the past several years. Future growth is expected with the emergence of high bandwidth applications, such as Tele-Immersion, HDTV, and broadband video to name a few. There has been growing interest in the realization of an optical Internet (IP over WDM) to support these future applications. The advantage of an optical Internet is that it eliminates the complexity and overhead associated with the ATM and SONET layers. Optical burst switching (OBS) is one approach that can be used to efficiently transmit bursts of IP data over WDM networks. In an OBS network, a WDM link consists of multiple data channels to transmit the payload (data bursts) and one or more dedicated control channels to transmit the corresponding burst header packets (BHP). The BHP is transmitted ahead of the burst in order to configure the switches along the burst's route. The burst follows the header without waiting for an acknowledgment for the connection establishment. The header and the data burst are separated at the source, as well as subsequent intermediate nodes, by an offset time. The offset time allows for the header to be processed at each node before the arrival of the burst; thus, no fiber delay lines are necessary at the intermediate nodes to delay the burst while the header is being processed. The control message may also specify the duration of the burst in order to let a node know when it may reconfigure its switch for the next burst, a technique known as *Just-Enough-Time* (JET) [1]. In this paper, we will consider an optical burst switched network which uses the JET technique.

One of the key issues in OBS is the scheduling of bursts on output data channels. Currently, contention resolution techniques used in scheduling are wavelength conversion and buffering. In wavelength conversion, if multiple bursts on the same wavelength compete for the same output port at the same time, then the bursts are shifted to different independent wavelengths. In optical buffering, FDLs are used to provide limited delay to the data bursts, proportional to the length of fiber delay line, in order to resolve the contention. Current data channel scheduling algorithms that use wavelength conversion and FDLs include latest available unscheduled channel (LAUC), and latest available unscheduled channel with void filling (LAUC-VF) [3]. However, these techniques drop the burst completely if all of the data channels are occupied at the arrival time of the burst. Instead of dropping the burst in its entirety, it is possible to drop only the overlapping parts of a burst. This technique is referred to as burst segmentation [2]. In this paper, we propose new scheduling techniques incorporating segmentation and FDLs. We show that the performance of the proposed algorithms in terms of packet loss probability is better than existing scheduling techniques.

This paper is organized as follows. Section II discusses the architecture of the FDLs. Section III discusses the new scheduling algorithms with burst segmentation and FDLs. Section IV provides numerical results. Section V concludes the paper and proposes directions for future research.

II. ARCHITECTURE OF OPTICAL ROUTERS

We consider two FDL architectures in the OBS node, for realizing the proposed scheduling algorithms to be described in the next section. The architecture in Fig. 1(a) shows an input-buffered FDL OBS node with FDLs dedicated to each input port, while Fig. 1(b) shows an output buffered FDL OBS node with FDLs dedicated to each output port. The basic architecture of the FDL buffered OBS node is introduced in [3].

In the input-buffer OBS node architecture shown in Fig. 1(a), each input port is equipped with an FDL buffer containing N delay lines. The n data channels are de-multiplexed from each input fiber link and are passed through wavelength converters whose function is to convert the input wavelengths to wavelengths that are used within the FDL buffers. The use of different wavelengths in the FDL buffers and on the output links helps to resolve contentions among multiple incoming data bursts competing for the same FDL and the same output link. In the design of FDL buffers, we can have fixed delay FDL buffers, variable delay FDL buffers, or a mixture of both. In this paper, we follow the architecture of variable delay FDL buffers.

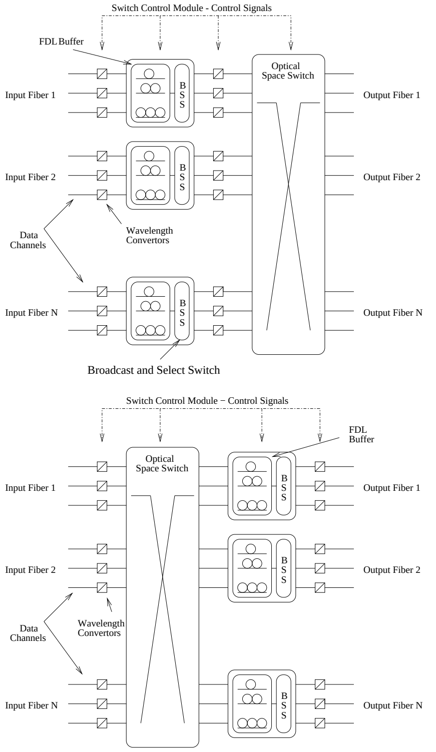

Fig. 1. (a) Input-Buffer and (b) Output-Buffer FDL Architecture.

In the output-buffer OBS node architecture, shown in Fig. 1(b), the FDL buffers are placed after the switch fabric. The input wavelength converters are used to convert the input wavelengths to the wavelengths that are used within the switching fabric. The functions of the output wavelength converters are the same as described in the input-buffer FDL architecture.

III. CHANNEL SCHEDULING ALGORITHMS

There has been substantial work on scheduling using FDLs in OBS [3]. In this section, we propose a number of scheduling algorithms incorporating both segmentation and FDLs. Based on the two FDL architectures presented in the previous section, we have two classes of scheduling schemes. Algorithms based on the input-buffer FDL node architecture are called *delay-first* scheduling algorithms, while scheduling algorithms based on the output-buffer FDL node architecture are called *segment-first* scheduling algorithms. In both schemes, we assume that full wavelength conversion, FDLs, and segmentation techniques are used to resolve burst contention for an output data channel. However, the order of applying the above techniques depends on the FDL architecture. In the delay-first scheme, we resolve contention by wavelength conversion, FDLs, and segmentation, in that order. In the segment-first scheme, we resolve contention by wavelength conversion, segmentation, and FDLs, in that order. Before going on to the detailed description of the schemes, it is necessary to discuss the motivation for developing two different schemes. In delay-first schemes, FDLs are primarily used to delay the entire burst, while in segment-first schemes, FDLs are primarily used to delay the segmented bursts. Delaying the entire burst and then segmenting the burst keeps the packets in order; however, when delaying segmented bursts, packet order is not always maintained. In general, segment-first schemes will incur lower delays than delay-first schemes.

We will now describe the new scheduling algorithms which use segmentation, wavelength conversion, and FDLs. In this paper, the bursts which have already been assigned a data channel are referred as the *scheduled bursts*, and the burst which arrives to the node waiting to be scheduled is referred to as the *unscheduled burst*.

When a BHP arrives at a node, a data channel scheduling algorithm is employed to assign a data channel on the outgoing link for the unscheduled burst. The channel scheduler obtains the burst arrival time and duration of the unscheduled burst from the BHP. The algorithm may need to maintain the latest available unscheduled time (LAUT), gaps, and voids, on every outgoing data channel. The LAUT of a data channel is the earliest time at which the data channel is available for an unscheduled data burst to be scheduled. A gap is the time between the arrival of the unscheduled burst and ending time of the previously scheduled burst. A void is the unscheduled duration between two scheduled bursts on a data channel. For void filling algorithms, the starting and the ending time for each burst on every data channel must be maintained.

The following information is needed by the scheduling algorithm:

- W: Maximum number of outgoing data channels.
- N_b: Maximum number of data burst scheduled on a data channel.
- MAX_DELAY: Maximum delay provided by FDL.
- $LAUT_i$: LAUT of the i^{th} data channel, $i = 1, 2, ..., W$.
- t_{ub}: Unscheduled burst arrival time.
- $S_{(i,j)}$ and $E_{(i,j)}$: Starting and ending times of each scheduled burst, j, on every data channel, i.
- L_b: Unscheduled burst length duration.
- D_i: i^{th} outgoing data channel.
- $Overlap_i$: Duration of overlap between the unscheduled burst and scheduled burst(s). Overlap is used in non-void filling channel scheduling algorithms. The overlap is zero if the channel is available, otherwise the overlap is the difference between $LAUT_i$ and t_{ub}.
- Gap_i: If the channel is available, gap is the difference between t_{ub} and $LAUT_i$ for scheduling algorithms without void filling, and is the difference between t_{ub} and $E_{(i,j)}$ of previous scheduled burst, j, for scheduling algorithms with void filling. If the channel is busy, Gap_i is set to 0. Gap may be used to select a channel for the case in which more than one channel is free.
- $Loss_i$: Total packet loss on i^{th} data channel, due to the assignment of the unscheduled burst. The primary goal of all scheduling algorithms is to minimize loss; hence, loss is the primary factor for choosing a data channel. In case the loss on more than one channel is the same, then other channel

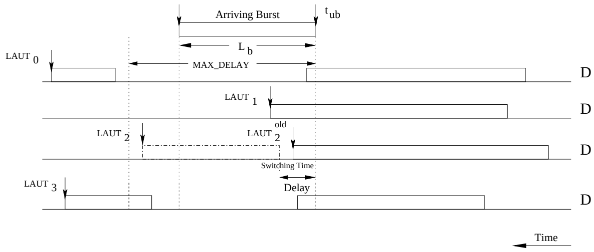

Fig. 2. Illustration of DFMOC.

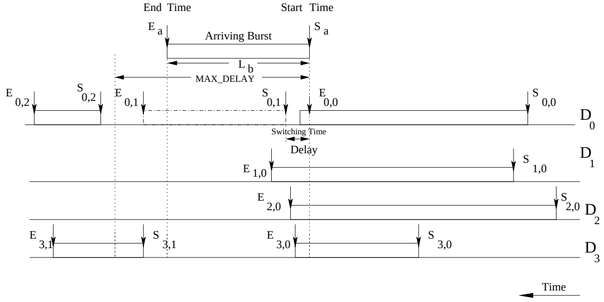

Fig. 3. Illustration of DFMOC-VF.

parameters are used to reach a decision on the selection of a data channel.

- $Void_{(i,k)}$: Duration of k^{th} void on i^{th} data channel. This information is relevant to void filling algorithms. A void is the duration between the $S_{(i,j+1)}$ and $E_{(i,j)}$ on a data channel. Void information is useful in selecting a data channel in case more than one channel is free.

Data channel scheduling algorithms can be broadly classified into two categories: without and with void filling. The algorithms primarily differ based on the type and amount of state information about every channel that are maintained at a node. In data channel scheduling algorithms without void filling, the $LAUT_i$ on every data channel D_i, $i = 0, 1, ..., W$, is maintained by the channel scheduler. In void filling algorithms, the starting time, $S_{(i,j)}$ and ending time, $E_{(i,j)}$ are maintained for each burst on every data channel, where, $i = 0, 1, ..., W$, is the i^{th} data channel and $j = 0, 1, ..., N_b$, is the j^{th} burst on channel i.

The data channel scheduling algorithms can be further classified into two categories: *Delay-First* and *Segment-First*.

A. Delay-First Channel Scheduling Algorithms

1. *Delay-First Minimum Overlap Channel (DFMOC):* The DFMOC algorithm calculates the overlap on every channel and then selects the channel with minimum overlap. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the minimum overlap is greater than or equal to the sum of the unscheduled burst length and MAX_DELAY, the entire unscheduled burst is dropped. Otherwise, the unscheduled burst is delayed for the duration of the minimum overlap and scheduled on the selected channel. In case the minimum overlap is greater than MAX_DELAY, the unscheduled burst is delayed for MAX_DELAY and the non-overlapping burst segment of the unscheduled burst is scheduled, while the overlapping burst segment is dropped. For example, in Fig. 2, the data channel D_2 has the minimum overlap, thus the unscheduled burst is scheduled on D_2 after providing a delay using FDLs.

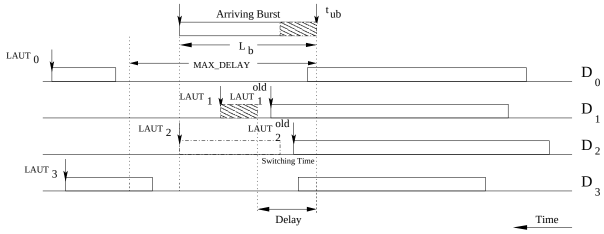

Fig. 4. Illustration of SFMOC.

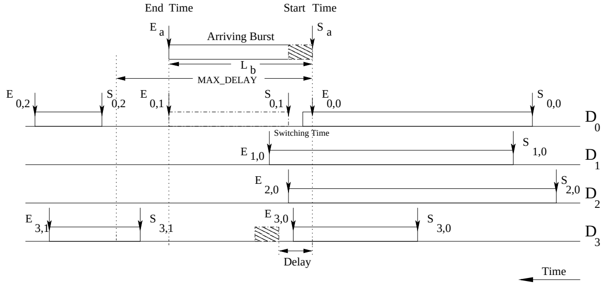

Fig. 5. Illustration of SFMOC-VF.

2. *Delay-First Minimum Overlap Channel with Void Filling (DFMOC-VF):* The DFMOC-VF algorithm calculates the delay until the first void on every channel and then selects the channel with minimum delay. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the starting time of the first void is greater than or equal to the sum of the end time, E_a, of the unscheduled burst and MAX_DELAY, the entire unscheduled burst is dropped. Otherwise, the unscheduled burst is delayed until the start of the first void on the selected channel, where the non-overlapping burst segment of the unscheduled burst is scheduled, while the overlapping burst segment is dropped. In case the start of the first void is greater than the sum of the start time, S_a, of the unscheduled burst and MAX_DELAY, the unscheduled burst is delayed for MAX_DELAY and the non-overlapping burst segment of the unscheduled burst is scheduled, while the overlapping burst segment is dropped. For example, consider Fig. 3, by applying the algorithm, the data channel D_0 has the minimum delay, thus the unscheduled burst is scheduled on D_0 after delaying it using FDLs. Here only the overlapping segments of the burst are dropped instead of the entire burst as in the case of LAUC-VF.

B. Segment-First Channel Scheduling Algorithms

1. *Segment-First Minimum Overlap Channel (SFMOC):* The SFMOC algorithm calculates the overlap on every channel and then selects the data channel with minimum overlap. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the minimum overlap is greater than or equal to the sum of the

unscheduled burst length and MAX_DELAY, the entire unscheduled burst is dropped. Otherwise, the unscheduled burst is segmented (if necessary) and the non-overlapping burst segment is scheduled on the selected channel, while the overlapping burst segment is re-scheduled. Now, the algorithm calculates the overlap on all the channels for the re-scheduled burst segment. The re-scheduled burst segment is delayed for the duration of the minimum overlap and scheduled on the selected channel. In case the minimum overlap is greater than MAX_DELAY, the re-scheduled burst segment is delayed for MAX_DELAY and the non-overlapping burst segment of the re-scheduled burst segment is scheduled, while the overlapping burst segment is dropped. For example, in Fig. 4, we observe that the data channel D_2 has the minimum overlap for the unscheduled burst, thus the unscheduled burst is scheduled on D_2, and the re-scheduled burst segment is scheduled on D_1.

2. *Segment-First Minimum Overlap Channel with Void Filling (SFMOC-VF):* The SFMOC-VF algorithm calculates the loss on every channel and then selects the channel with minimum loss. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the starting time of the first void is greater than or equal to the sum of the end time, E_a, of the unscheduled burst and MAX_DELAY, the entire unscheduled burst is dropped. If the starting time of the first void is greater than or equal to the end time, E_a, of the unscheduled burst, the DFMOC-VF algorithm is employed. Otherwise, the unscheduled burst is segmented (if necessary) and the non-overlapping burst segment is scheduled on the selected channel, while the overlapping burst segment is re-scheduled. For the re-scheduled segment, the algorithm calculates the delay required until the start of the next void on every channel and selects the channel with minimum delay. The re-scheduled burst segment is delayed until the start of the first void on the selected channel and the non-overlapping burst segment of the re-scheduled burst is scheduled, while the overlapping burst segment is dropped. In case the start of the next void is greater than the sum of the start time, S_a, of the unscheduled burst and MAX_DELAY, the re-scheduled burst segment is delayed for MAX_DELAY and the non-overlapping burst segment of the re-scheduled burst is scheduled, while the overlapping burst segment is dropped. For example, in Fig. 5, we observe that the data channel D_0 has the minimum loss, thus the unscheduled burst is scheduled on D_0, and the unscheduled segmented burst is scheduled on D_3 (as it gives the minimum delay) after providing a delay using FDLs.

Table 1 compares all the above discussed data channel scheduling algorithms in terms of time complexity and the amount of state information stored. We can observe that the time complexity of the non-void filling algorithms is less than the void filling algorithms. Also, void filling algorithms, such as, LAUC-VF, DFMOC-VF, and SFMOC-VF, store more state information as compared to non-void filling algorithms, such as, LAUC, DFMOC, and SFMOC.

TABLE I

Comparison of Channel Scheduling Algorithms

No.	Algorithm	Time Complexity	State Information
1	LAUC	O(W)	$LAUT_i$, Gap_i
2	LAUC-VF	O(W$\cdot \log N_b$)	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i
3	DFMOC	O(W)	$LAUT_i$, Gap_i
4	DFMOC-VF	O(W$\cdot \log N_b$)	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i
5	SFMOC	O(2·W)	$LAUT_i$, Gap_i
6	SFMOC-VF	O(2·W$\cdot \log N_b$)	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i

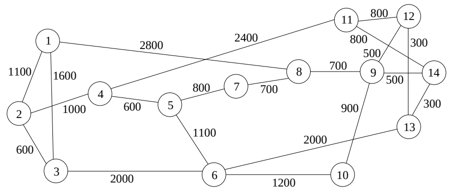

Fig. 6. 14-Node NSF Network.

IV. Numerical Results

In order to evaluate the performance of the new data channel scheduling algorithms, a simulation model is developed. Burst arrivals to the network are Poisson with rate λ. Burst length is an exponentially generated random number rounded to the nearest integer multiple of the fixed sized packet length. Mean burst length, $\frac{1}{\mu}$, is 100 ms, load is measured in Erlang, and transmission rate is 10 Gbps. Packet length is 1250 bytes. Switching time is 10 μs. All bursts have the same initial offset time of 20 μs, with burst header processing time at each node being 2.5 μs. Dijkstra shortest-path routing algorithm is used to find the path between all node pairs. All the simulation results are obtained for an 14-node NSF network (Fig. 6) with 8 data channels on each link. Figure 7 plots the total packet loss probability versus load for different channel scheduling algorithms. We observe that the channel scheduling algorithms with burst segmentation perform better than algorithms without burst segmentation at most loads. Also, the delay-first algorithms have lower loss as compared to the segment-first algorithms. This is due to the possible blocking of the re-scheduled burst seg-

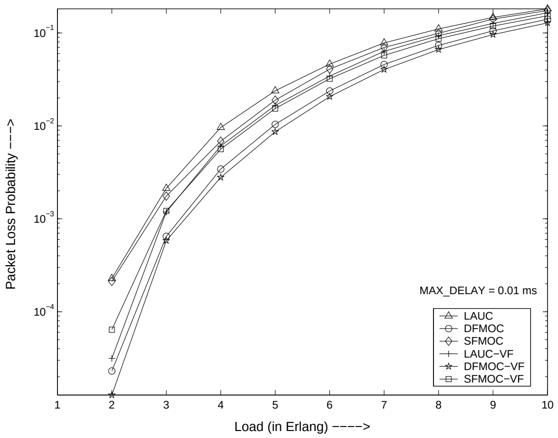

Fig. 7. Packet loss probability versus load for different scheduling algorithms with number of data channels, $D = 8$.

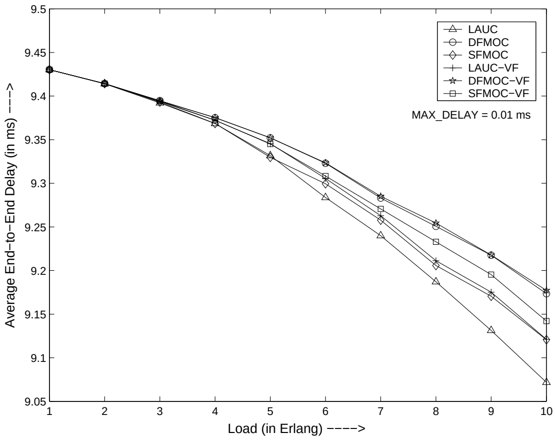

Fig. 8. Average end-to-end delay versus load for different scheduling algorithms with number of data channels, $D = 8$.

ment by the recently scheduled non-overlapping burst segment in the segment-first algorithms. The loss obtained by delay-first algorithms is the lower bound for the segment-first algorithms. We observe that at any given load, the DFMOC and DFMOC-VF algorithms perform the best, since the unscheduled burst is delayed and in case there is still a contention, the burst is segmented and only the overlapping burst segment is dropped. The segment-first algorithms lose packets proportional to the switching time, everytime there is a contention, while the LAUC and LAUC-VF algorithms delay the burst in case of a contention and schedule the burst if the channel is free after the provided delay. Hence, at low loads LAUC-VF performs better than SFMOC-VF, and as the load increases SFMOC-VF performs better. Therefore a substantial gain is achieved by using segmentation and FDLs.

Figure 8 plots the average end-to-end delay versus load for different channel scheduling algorithm. We observe that the delay-first algorithms have higher delay as compared to the segment-first algorithms, since FDL is the primary contention resolution technique in the former and segmentation is the primary contention resolution technique in the latter. The higher delay for scheduling algorithms with segmentation is due to the higher probability of a successful transmission between source-destination pairs which are further apart. In traditional scheduling algorithms, the entire burst is dropped in case of a contention; hence, only source-destination pairs close to each other have a higher probability of making a successful transmission, which results in lower average end-to-end packet delay. Hence, we can carefully choose either delay-first or segment-first schemes based on loss and delay tolerances of input IP packets.

Figure 9 plots the packet loss probability versus load for different channel scheduling algorithm for different MAX_DELAY values. When MAX_DELAY is set to zero, i.e., no FDLs are used, algorithms with segmentation outperform the traditional algorithms. When a high MAX_DELAY value of 0.05 ms is used, algorithms which use FDLs as the primary contention resolution technique, such as, LAUC, LAUC-VF, DFMOC, DFMOC-VF outperform the algorithms which use segmentation as the primary contention resolution technique, such as, SFMOC, SFMOC-VF.

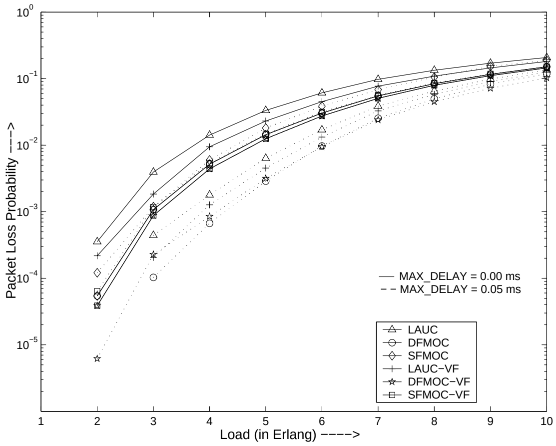

Fig. 9. Packet loss probability versus load for different scheduling algorithms with number of data channels, $D = 8$ for different MAX_DELAY values.

V. CONCLUSION

In this paper, we considered the concept of burst segmentation and FDLs for burst scheduling in optical burst-switched networks, and we have proposed a number of new channel scheduling algorithms for OBS networks. The scheduling algorithms with burst segmentation perform better than the existing scheduling algorithms, with and without void filling, in terms of packet loss. We also introduced two categories of scheduling algorithms based on the FDL architecture. The delay-first algorithms are suitable for transmitting packets which have higher delay tolerance and strict loss constraints, such as Internet data, while the segment-first algorithms are suitable for data with higher loss tolerance and strict delay constraints, such as voice and video traffic.

Areas of future works include extending the proposed algorithms to support QoS. In the case of providing QoS support, the priority of the burst can be stored in the BHP [5], and the scheduling algorithm can dynamically decide which of contending bursts or burst segments to drop using burst segmentation or to delay using FDLs. It would also be useful to develop an analytical model for the scheduling algorithms.

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] V.M. Vokkarane, J.P. Jue, and S. Sitaraman, "Burst Segmentation: an Approach for Reducing Packet Loss in Optical Burst Switched Networks," *Proceedings, IEEE ICC 2002*, New York, NY, vol. 5, pp. 2673-2677, April 2002.

[3] 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, October 2000.

[4] S. Yao, B. Mukherjee, S.J.B. Yoo, and S. Dixit,"All-Optical Packet-Switched Networks: A Study of Contention Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets," *Proceedings, SPIE OptiComm 2000*, Dallas, TX, pp.235-246, October 2000.

[5] V.M. Vokkarane and J.P. Jue, "Prioritized Routing and Burst Segmentation for QoS in Optical Burst-Switched Networks," *Proceedings, IEEE/OSA OFC 2002*, Anaheim, CA, WG6, pp. 221-222, March 2002.