Burst Scheduling Based on Time-slotting and Fragmentation in WDM Optical Burst Switched Networks

G. Mohan*, M. Ashish, and K. Akash
Department of Electrical and Computer Engineering
National University of Singapore, Singapore 117576
Email: elegm@nus.edu.sg

ABSTRACT
The recent explosive growth in the Internet technology has made it imperative to have a network which supports a very high bandwidth.*Wavelength division multiplexing (WDM)* based on optical networks are becoming a viable solution to meet this ever-increasing bandwidth demand. *Optical burst switched (OBS)* networks provide a good solution for a very high traffic influx which is generally bursty in nature. In this a single control header is sent for a group of packets which constitute a burst. A good scheduling algorithm which provides efficient resource utilization and maximizes acceptance ratio for bursts is desired. At the same time it is important that the algorithm is not very computationally intensive. In this paper we develop a computationally simple algorithm which schedules bursts in a more efficient way than the existing algorithms [1], [2]. The proposed algorithm uses the concepts of *time-slotting* and *burst fragmentation* to improve performance in terms of burst dropping probability. Extensive simulation results have been used to demonstrate the performance of the algorithm.

KEY WORDS
IP over WDM, Wavelength division multiplexing, Optical burst switching, Scheduling, Time-slotting, Burst fragmentation.

1 Introduction

Potential bottlenecks of electronic processing to carry IP packet traffic over WDM optical networks can be overcome by aggregating multiple data packets into a super packet, called burst. Bursts are a collection of data packets assembled at an ingress router, having the same network egress address and some common attributes, like QoS requirements [1], [2]. A network which supports this kind of switching is called an optical burst switched (OBS) network. The basic difference between OBS and circuit switching is that in burst switching, bandwidth is reserved in a one-way process; that is, a burst can be sent out without waiting for the acknowledgement of a successful reservation, whereas in circuit switching, bandwidth is reserved in a two-way process. In other words, data can only be sent after a circuit has been successfully established. This results in a longer latency. As compared to optical packet switching, a burst cuts through the intermediate nodes without being buffered, whereas in packet switching, packets are optically stored and forwarded at each intermediate node, leading to increased processing complexity, buffering, and synchronization problems [3], [4], [5].

A burst consists of a burst header and a burst payload. The burst payload is also called data burst. In an OBS network, the data burst and its header are transmitted separately with a time lag. This time lag is called the offset time. The minimum offset time is given by $h\Delta$, where Δ is the processing time of the control header at each node along the path with h hops [3, 4]. This processing time is the difference between the time when the control header enters a node and the time when it leaves the node after scheduling all the resources. The data burst and the control header are sent on different wavelengths/channels with the burst header slightly ahead in time, and are switched in optical and electronic domains, respectively, at each core router they traverse. The burst header contains all the necessary routing information to be used by the switch control unit (SCU) at each hop to schedule the data burst and configure the optical switching matrix to switch the data burst optically [1], [3]. The separate transmission and switching of data bursts and their headers help to facilitate the electronic processing of headers and lower the opto-electronic processing capacity required at core routers. Further, it provides ingress to egress transparent optical paths for transporting data bursts.

In the event of no wavelength being available at the time of arrival of a burst at a node, the burst is dropped. In order to minimize this an efficient burst scheduling (wavelength channel scheduling) algorithm is required to choose the best wavelength on the outgoing link for the data burst. In the literature, quite a few of these scheduling algorithms have been proposed [1], [2], [6]. These include, First Fit Unscheduled Channel (FFUC), Latest Available Unscheduled Channel (LAUC) and Latest Available Void Filling Algorithm (LAVF). Among these three algorithms LAVF has the best performance and leads to maximum efficiency, but this is achieved at the cost of a high time complexity. The other two algorithms strike a compromise between efficient resource utilization and time complexity.

In this paper we propose an algorithm called *Best fit*

* Author for correspondence

void filling with fragmentation (BFVFF). The benefits of this algorithm is two-fold. BFVFF results in a very efficient resource utilization which in turn leads to a relatively higher acceptance ratio and at the same time its time complexity is very low compared to the other existing algorithms [1], [2]. The algorithm is based on the two key concepts of time slotting and fragmentation. The claims are substantiated with extensive simulation experiments.

The rest of the paper is organized as follows. In Section 2, related work on burst switching protocols and scheduling is discussed, followed by the details of the proposed burst scheduling technique in Section 3. Section 4 presents the results of performance study while Section 5 makes some concluding remarks.

2 Related Work

In the literature several optical burst switching protocols have been proposed. Among these protocols, Just-Enough-Time (JET) which is based on Reserve-a-Fixed-Duration (RFD) is very attractive [3], [4]. Various algorithms for scheduling bursts have been proposed in the literature[1], [2]. These algorithms differ in their burst dropping performance and complexity. The algorithms which provide efficient resource utilization suffer in computational complexity while those that have low computational complexity compromise on the performance. This is the main motivation behind the research, to develop an algorithm which provides efficient resource utilization and has low time complexity.

In this section a close comparison is made between these algorithms and BFVFF algorithm developed by us.

FFUC Algorithm: In the FFUC Algorithm the unscheduled time is maintained for each of the outgoing wavelength channels. A channel is eligible for allocation only if its unscheduled time is less than the burst arrival time. This algorithm has a worst case time complexity of $O(W)$ where W is the number of wavelengths. BFVFF algorithm also has a worst case time complexity of $O(W)$, but compared to FFUC it offers a far better link utilization. This is mainly because FFUC leads to the creation of many voids.

LAUC Algorithm: Like the FFUC algorithm, the LAUC algorithm also follows the same criterion for eligibility of a channel for scheduling a burst. The difference however lies in the fact that LAUC tries to select the wavelength for which the unscheduled time is closest to the burst arrival time. This reduces the void size and improves performance. The time complexity of this algorithm is also $O(W)$, but the acceptance ratio is significantly lower than LAVF and also BFVFF algorithm.

LAVF Algorithm: The LAVF algorithm maintains information of every burst scheduled on an outgoing channel. Using this it keeps track of all the voids created in the time domain. LAUC considers a channel eligible for allocation only if the channel is unscheduled for the entire burst duration. Among all the eligible channels, the algorithm selects the latest available one. This uses the voids created effectively. The time complexity for the algorithm is $O(WN)$ where, N is the total number of voids or bursts scheduled. However, this time complexity can be reduced to $O(WlogN)$ by using complex data structures. Among the three algorithms mentioned LAVF gives the best results, though its time complexity is comparatively higher. On comparison BFVFF gives an even better resource utilization than LAVF and its time complexity is also lower.

3 Proposed Algorithm

In the conventional burst scheduling algorithms like LAVF, each wavelength on a fiber is considered as a separate resource and a burst can be scheduled on only one of them at a time. This however leads to wastage on each of the wavelengths and the cumulative wastage on all the wavelengths is quite significant. If all the wavelengths on a given fiber can be treated as a single resource then this wastage can be minimized. This is the motivation for the proposed algorithm. BFVFF (Best fit void filling with fragmentation) is based on two fundamental principles, time-slotting and burst fragmentation. Time-slotting refers to the quantization of time into slots of a fixed size. Each node in the network has such a time-line for all the wavelengths on each of the outgoing links. This is used to schedule the outgoing bursts on a specific wavelength on a specific link. The time-slotting being referred to here is fundamentally different from the photonic slots used in the *photonic slot routing in WDM packet-switched networks* [7], [8].

Burst fragmentation refers to the splitting of the burst at a node if it cannot be accommodated as a whole, on any of the wavelengths of the particular link it is supposed to go on. This fragmentation can be iterative and each fragment of the burst can be fragmented again till a minimum permissible size is reached. However, this is flexible and a lower limit for the fragment size can be set as a multiple of time slots. Experimental results have been taken for this case as well and the effects of changing this lower limit are discussed later. The features of the algorithm are explained in more detail in the subsections that follow.

3.1 Time-slotting Approach

Time-slotting helps in improving the computational complexity of a void filling algorithm. This is because the burst arrival time at a node is known while scheduling the burst. This arrival time can be used to directly reference the time-line of the resource being allocated, and we can determine if the resource is free at that time. No extensive search needs to be performed on the resource schedule. Thus if there are W wavelengths on a particular link then the worst case time complexity for searching for space to schedule the burst is $O(kW)$ where, k is the maximum number of time slots a burst can span. As this is a constant factor, the worst case time complexity becomes $O(W)$. When used

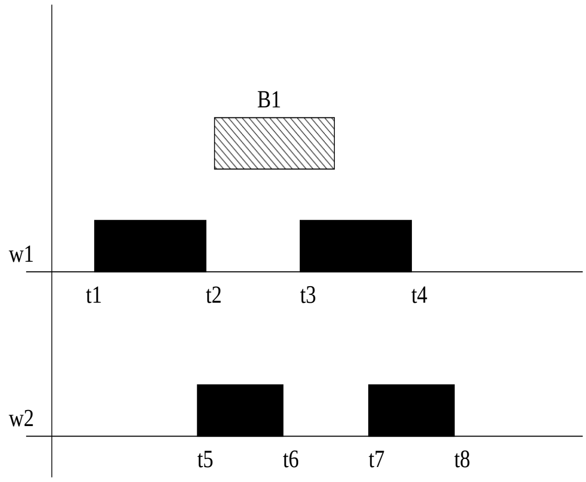

Figure 1. A situation wherein LAVF fails to schedule burst B1

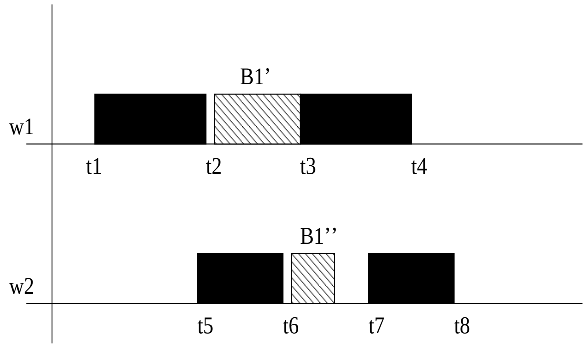

Figure 2. Burst B1 is scheduled by using fragmenting over w_1 and w_2.

with burst fragmentation, the burst can be scheduled on one or more wavelengths as long as required slots are available. For instance, a burst may be scheduled on wavelength w_1 at slot t_1, then a different wavelength w_3 during slot t_2.

The size of the time slot can be determined by various factors such as mean burst duration, network properties and statistics about the flow of data. For good performance of the algorithm, the time slot should be small compared to the burst duration. If the average size of a burst is about 25μs, we could divide the time in 1μs divisions. The effect of varying time slot size on the performance of the algorithm is discussed in the next section on performance study.

As an integral number of time slots is allocated to a burst whose starting and ending times may not be necessarily aligned to the beginning and ending of a time slot respectively, an average of half a time slot is wasted on either end of the burst. Alternatively, the number of slots wasted in this process can be reduced by aligning the beginning of the burst with the beginning of a time slot. This can be achieved by delaying the burst at a node by using fiber delay lines (FDLs) to make link propagation delay a multiple of time slot. If a 1 μs time slot is used, maximum length of FDL needed is 300m for any link.

3.2 Fragmentation Approach

Burst fragmentation helps in increasing the acceptance ratio for a scheduling algorithm. For example in Fig. 1, if we use LAVF, the new burst B1 cannot be scheduled on either wavelength w_1 or w_2 and hence will be dropped. However if we allow burst fragmentation, as shown in Fig. 2, then the burst can be accommodated on wavelengths w_1 and w_2 by fragmenting the burst at time t_3 and scheduling the first fragment B1' on w_1 and the second fragment B1" on wavelength w_2. As long as a free time slot can be found for each slot of the burst duration on any of the wavelengths, the burst will not be dropped. This improves the burst acceptance ratio.

When a burst needs to be fragmented at a node its control header is updated and each fragment of the burst is treated like an independent burst, thus enabling further fragmentation. Final assembly of the burst is done at the egress. This is possible because it will be known at the egress, which fragment comes at what time and which wavelength.

In order to take care of the issues that arise on implementation, guard bits need to be inserted in the burst. This is to account for the time which is required by the control circuitry to configure so that the subsequent bits are sent to another wavelength. This guard time is a very small percent of the burst duration and hence does not significantly add on to the burst size. We note that even without fragmentation, some time gap is needed when two bursts are scheduled in continuation. The guard bits need to be inserted at every possible point where the burst can be fragmented. This is determined by the lower limit of fragment size. This technique does not require the use of FDLs.

3.3 BFVFF Scheduling Algorithm

When a control packet arrives at a node, the burst duration b measured in slots and the starting time slot t_1 are first determined. The scheduling algorithm is then used to assign wavelengths to the burst. In the pseudo-code given below, f denotes the minimum fragment size and w_{max} denotes the wavelength with the maximum number of free contiguous time slots starting from some time t.

$l = b$; $t = t_1$;
while $(l > 0)$
    Determine w_{max} starting from t;
    $max =$ no. of free contiguous slots on w_{max};
    if $(max < l)$
        if $(max < f)$ drop the burst, break;
        $max = \lfloor \frac{max}{f} \rfloor * f$;
    Schedule the burst on w_{max} for max slots;
    $l = l - max$; $t = t + max$;

The maximum number of slots scanned per wavelength in each iteration is max and the next iteration only starts from time $t + max$, thus at most b slots are searched on each wavelength leading to the $O(W)$ worst case time complexity since b is bound by the maximum burst size.

4 Performance Study

We study the performance of the BBVFF algorithm through simulation on a random network with 32 nodes and 104 links. Each link is assumed to carry 8 wavelengths. Bursts arrive randomly according to poisson process, with negative exponentially distributed duration with a mean of 25μs. The destination nodes for the bursts are generated using a uniform distribution. The arrival rate used here is the mean number of bursts generated per node, per mean burst duration. Control processing time is assumed to be 20μs and the offset factor is taken to be 1, unless mentioned otherwise. Offset factor refers to a multiple of the basic offset value $h\Delta$. The time slot size is chosen to be 1μs for most of the simulations, however, we do analyse the effect of varying slot size. The simulation experiments were run for a sufficiently long time and were repeated several times to get accurate values with 95% confidence interval. The performance metric used in the experiments is acceptance ratio (or equivalently dropping probability) which is the ratio of bursts successfully scheduled to the total number of bursts generated.

4.1 Effect of Traffic Load

We start with comparing the acceptance ratio for different arrival rates while varying the fragment size. When no burst fragmentation is allowed the algorithm starts behaving like LAVF and the performance thus follows that of LAVF very closely. As can be seen from the Fig. 3, when the burst is allowed to fragment the acceptance ratio improves significantly. For instance, for the arrival rate 20, 7% bursts which are dropped in the no fragmentation case are accepted when completely fragmented. Similar behavior can be observed for other arrival rates as well. A compromise between the two scenarios can also be seen when the burst is allowed to fragment only after every 5 time slots.

4.2 Effect of Fragment Size

In Fig. 4, the effect of fragment size becomes even more apparent. As the permissible fragment size is reduced the performance improves, thus giving the best results when the burst is allowed to fragment at every time slot. This is because of the fact that with the decrease in the smallest fragment size, the size and the number of voids created decreases. However the benefit of having a larger fragment size is that the number of guard bits are reduced and the average number of fragments for each burst reduces. This in turn reduces the control header processing time. The algorithm causes an improvement in all the arrival rates, as is evident from the graph.

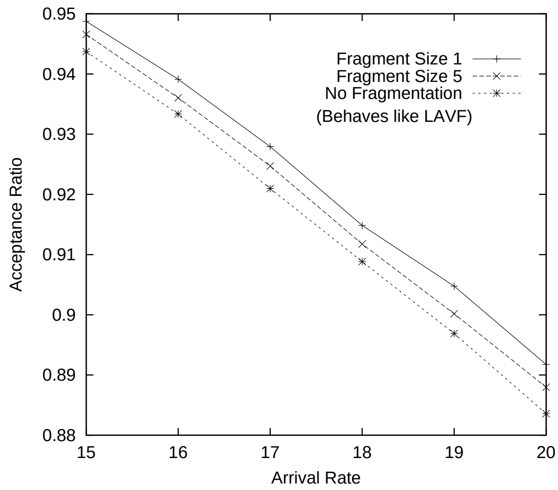

Figure 3. Acceptance Ratio vs Arrival Rate for different levels of fragmentation.

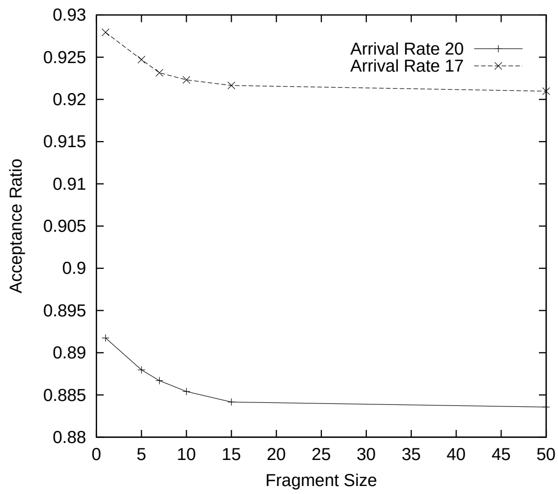

Figure 4. Effect of Fragment Size on Dropping Probability for different Arrival Rates.

4.3 Effect of Slot Size

The effect of varying the time slot size was also studied. As we have mentioned earlier, discretization of the time domain results in a waste of link bandwidth for half the size of time slot, statistically. Thus, as we increase the size of time slot, wastage increases and hence fewer bursts can be scheduled over the network. Fig. 5 shows how the performance deteriorates with increasing time slot size. A smaller slot size is always better, but it requires larger memory space for storing the scheduling status and hence a larger constant processing time.

4.4 Effect of Offset Factor

The effect of changing offset factor is depicted in Fig. 6. When the offset factor is increased the acceptance ratio in-

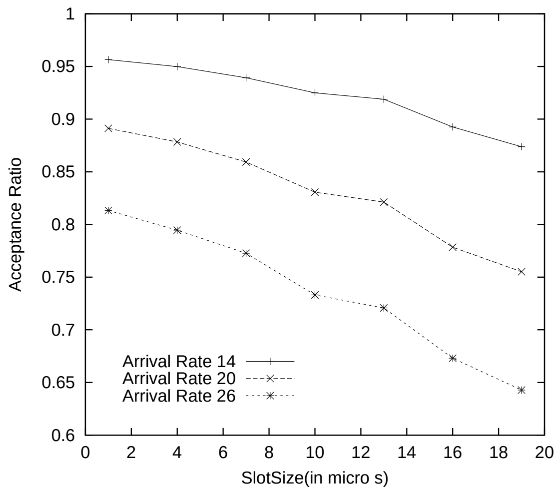

Figure 5. Effect of slotsize on the acceptance ratio

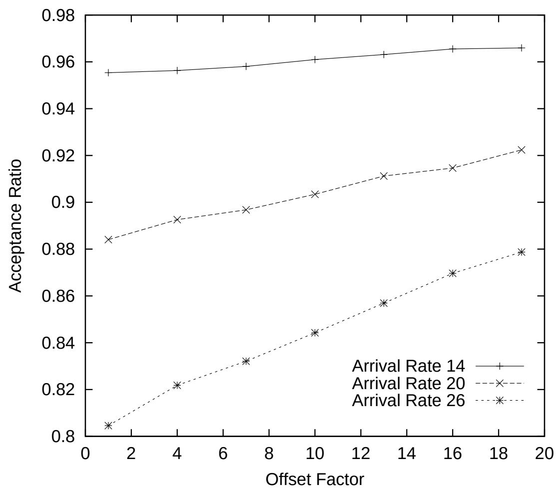

Figure 6. Effect of Offset Factor on the Acceptance Ratio

creases. This is because the bursts get scheduled in advance, leading to a better scheduling.

5 Conclusions

In this paper, we have proposed a new algorithm for burst scheduling in optical burst switched networks. As compared to existing algorithms such as LAVF, our algorithm yields better performance in terms of resource utilization and computational complexity. In the algorithm, low computational complexity has been achieved by using time-slotting, and an efficient resource utilization has been achieved by using burst fragmentation. The implementation issues for the algorithm have also been discussed. Although having a small time-slot size maximizes resource utilization, it cannot be made very small because the constant processing time for the algorithm increases. This is because of the increase in the constant factor k in $O(kW)$. Similarly, increased fragmentation increases resource utilization, but there is an upper limit to fragmenting the burst. This occurs when the amount of guard bits become significant as compared to the fragment size. Greater fragmentation also adds to the control header, thus increasing control processing time. A good compromise for these factors has been suggested to give optimal performance. The effectiveness of our algorithm has been verified by extensive simulation experiments.

References

[1] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control Architecture in Optical Burst-Switched WDM networks" *IEEE Journal on Selected Areas in Communications*, pp. 1838-1851, October 2000.

[2] J. S. Tumer, "Terabit Burst Switching," *Journal of High Speed Networks*, 1999.

[3] C. Qiao and M. Yoo, "Choices, Features and Issues in Optical Burst Switching", *Optical Networks Magazine*, pp. 36-44, April 2000.

[4] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration" *IEEE Communication Magazine*, pp. 104-114, September 2000.

[5] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for Optical Packet Switching and Optical Burst Switching" *IEEE Communication Magazine*, pp. 136-142, January 2001.

[6] C. Seva Ram Murthy and Mohan Guruswamy, "WDM Optical Networks: Concepts, Design and Algorithms" *Prentice Hall, NJ, USA*, November 2001.

[7] H. Zang, J. P. Zue, and B. Mukherjee, "Capacity Allocation and Contention Resolution in a Photonic Slot Routing All-Optical WDM Mesh Network" *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1728-1741, December 2000.

[8] I. Chlamtac, V. Elek, A. Fumagalli, and C. Szabo, "Scalable WDM access network architecture based on photonic slot routing," *IEEE/ACM Transactions on Networking*, pp. 1-9, Feb 1999.