

Master's thesis

Title

**On Packet Scheduling Algorithm for WDM-based Photonic Packet
Switch with Fiber Delay Line Buffers**

Supervisor

Prof. Masayuki Murata

Author

Takashi Yamaguchi

February 12th. 2003

Department of Informatics and Mathematical Science

Graduate School of Engineering Science

Osaka University

Master's thesis

On Packet Scheduling Algorithm for WDM-based Photonic Packet Switch with Fiber Delay Line Buffers

Takashi Yamaguchi

Abstract

The progress in optical transmission technology has been remarkable, but the electronic technology for switching systems is approaching its limit. Therefore, in order to utilize broadband optical transmission, photonic packet switching in which packets are switched and buffered in an optical domain is required.

In this thesis, we comparatively evaluate two photonic packet switch architectures with WDM-based FDL buffers for synchronized variable-length packets. The first one is an output buffer type switch, which stores packets in the FDL buffer attached to each output port. The other is a shared buffer type switch, which stores packets in the shared FDL buffer. The performance of a switch is greatly influenced by its architecture and a packet scheduling algorithm. We compare the performances of these two packet switches by applying different packet scheduling algorithms. Through simulation experiments, we show that each architecture has a parameter region for achieving better performance. For the shared buffer type switch, we found that void space introduces unacceptable performance degradation when the traffic load is high. Accordingly, we propose a void space reduction method. Our simulation results show that our proposed method enables the shared buffer type switch to outperform the output buffer type switch even under high traffic load conditions. Next, we consider the feasible design of the architecture for the algorithms with PLD design software, and we discuss the feasibility of the algorithms from the viewpoint of processing delay time. We found that the number of wavelengths in a switch significantly influences processing delay time of packet scheduling algorithm. Thus, we proposed wavelength grouping method in order

to decrease the processing delay time which depends on the number of wavelengths. This method declines the performance of a switch. However, using our proposed algorithm, we can achieve the maximum performance of the switch when we decide the proper number of wavelengths according to the limitation of the hardware processing ability.

Keywords

WDM (Wavelength Division Multiplexing), Photonic Packet Switch, FDL (Fiber Delay Line) Buffer, Variable Length Packet, Packet Scheduling Algorithm, Hardware Feasibility

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 7 |
| 2 | Photonic Packet Switch Architectures | 10 |
| 3 | Packet Scheduling Algorithms | 15 |
| 3.1 | Buffer Control Algorithms | 15 |
| 3.2 | Void Space Reduction Method | 16 |
| 4 | Performance of the Photonic Packet Switches | 20 |
| 4.1 | Simulation Model | 20 |
| 4.2 | Evaluation of the Packet Scheduling Algorithms | 20 |
| 4.3 | Evaluation of Void Space Reduction Method | 28 |
| 4.4 | Effects of Increasing the Number of Wavelengths on FDLs | 29 |
| 5 | Feasibility of Packet Scheduling Algorithms | 31 |
| 5.1 | PLD Implementation and Simulation | 31 |
| 5.2 | Wavelength Grouping Method | 32 |
| 5.3 | Consideration of feasibility of packet scheduling algorithms | 35 |
| 6 | Conclusion | 38 |
| | Acknowledgements | 39 |
| | References | 40 |

List of Figures

| | | |
|----|--|----|
| 1 | Block diagram of photonic packet switch | 11 |
| 2 | Synchronization of packets inside a switch | 11 |
| 3 | Output buffer type photonic packet switch architecture | 12 |
| 4 | Shared buffer type photonic packet switch architecture | 13 |
| 5 | Void space in shared buffer type switch | 18 |
| 6 | Void space reduction method | 19 |
| 7 | Packet loss probability (output buffer type switch, load = 0.4) | 21 |
| 8 | Packet loss probability (shared buffer type switch, load = 0.4) | 21 |
| 9 | Packet loss probability (output buffer type switch, load = 0.6) | 22 |
| 10 | Packet loss probability (shared buffer type switch, load = 0.6) | 22 |
| 11 | Packet loss probability (output buffer type switch, load = 0.8) | 23 |
| 12 | Packet loss probability (shared buffer type switch, load = 0.8) | 23 |
| 13 | Packet loss probability (output buffer type switch, for $B = 64$) | 24 |
| 14 | Packet loss probability (shared buffer type switch, for $B = 64$) | 24 |
| 15 | Data loss probability (output buffer type switch, load = 0.4) | 25 |
| 16 | Data loss probability (shared buffer type switch, load = 0.4) | 25 |
| 17 | Data loss probability (output buffer type switch, load = 0.6) | 26 |
| 18 | Data loss probability (shared buffer type switch, load = 0.6) | 26 |
| 19 | Data loss probability (output buffer type switch, load = 0.8) | 27 |
| 20 | Data loss probability (shared buffer type switch, load = 0.8) | 27 |
| 21 | Packet loss probability (shared buffer type switch, void space reduction method, load = 0.6) | 29 |
| 22 | Packet loss probability (shared buffer type switch, increasing the inner wave- lengths, load = 0.6) | 30 |

| | | |
|----|---|----|
| 23 | Packet loss probability (shared buffer type switch, void space reduction method + increasing the inner wavelengths, load = 0.6) | 30 |
| 24 | Processing delay times for scheduling phases | 33 |
| 25 | Processing delay times for packet scheduling algorithms | 33 |
| 26 | An example of wavelength grouping method (4 × 4 switch, 8 wavelengths, 4 groups) | 34 |
| 27 | Packet loss probability (wavelength grouping method, algorithm A1 without void space reduction method, load = 0.6) | 36 |
| 28 | Packet loss probability (wavelength grouping method, algorithm A1 with void space reduction method, load = 0.6) | 36 |

List of Tables

| | | |
|---|---|----|
| 1 | Wavelength groups (4×4 switch, 8 wavelengths, 4 groups) | 35 |
| 2 | Technology road map for performance of ASIC [1] | 35 |

1 Introduction

The progress of optical transmission technology in recent years has been remarkable especially in achieving a Tbps class of transmission speed. However, as the bandwidth is increasing sharply because of advances in optical transmission technology, the electronic technology for switching systems is approaching its limit. Thus, we need a photonic network, which can incorporate functions such as the multiplexing, demultiplexing, switching, and routing functions in an optical domain, through which electronic control can be minimized. Then, we can expect to see a super-high speed network that exceeds the speed limit of the electronic devices.

In this thesis, we study packet scheduling algorithms for the photonic packet switch [2, 3]. In the packet switch, packet loss is caused by the contention of more than two packets destined for the same output port. In the conventional electronic switch, the output times of those packets are shifted by a store-and-forward technique utilizing RAM (Random Access Memory), and resolving packet contention is a simple procedure. However, in the photonic packet switch, we need to take other approaches because RAM in an optical domain is still not available. For instance, optical buffering is achieved by using optical fiber delay lines (FDL) for packet contention resolution [4, 5, 6, 7, 8, 9]. Using FDL, packets are stored in different lengths of delay lines, through which the departing times of packets are time-shifted. Another technique used for resolving packet contention is to introduce wavelength conversion on FDL, where the wavelengths of more than two packets contending the same output port are converted to different wavelengths by using tunable wavelength converters [10, 11, 12]. Although wavelength conversion requires a higher hardware cost, it results in a better performance [13, 14]. However, once the packet is injected into the FDL, it cannot be sent to the output port for the time duration corresponding to the length of FDL. Thus, we need an effective packet scheduling algorithm for WDM-based FDL (or WDM-FDL in short), and this is the main subject of this thesis.

In this thesis, we evaluate the performance of photonic packet switches with WDM-FDL sup-

porting variable-length packets. We assume that all arriving packets are synchronized at the pre-defined time slot, and packet length is given by an integer multiple of the time slot. Note that time-synchronization of asynchronously arriving packets can be realized by the technique presented in [15]. In this thesis, we consider two switching architectures. The first one is an output buffer type switch, which stores packets in the WDM-FDL buffer attached to each output port. The other is a shared buffer type switch, where all the packets failing to acquire the output port are sent to the single FDL buffer within the switch. As described above, the use of a packet scheduling algorithm is important for enabling the photonic packet switches to achieve a high performance. This is especially true for the shared buffer type architecture as we will show in a later section. We apply four packet scheduling algorithms proposed in [16, 17, 18] to the above two packet switching architectures and comparatively evaluate the performance of the switches. Also, we propose a new packet scheduling algorithm applicable to the shared buffer type switch, called the *void space reduction method*.

Furthermore, since our new packet scheduling algorithm is more complicated than other packet scheduling algorithms, we consider a feasible design for packet scheduling algorithms with PLD (Programmable Logic Device) design software. Through simulations, we measure processing delay time of packet scheduling algorithms. and propose the *wavelength grouping method* of decreasing the processing delay time for the packet scheduling algorithms. Finally, we discuss the feasibility of the algorithms from the viewpoint of processing delay time using the simulation results and a technology road map for semiconductors.

The rest of the thesis is organized as follows. In Section 2, we briefly present the shared buffer and output buffer type architectures for photonic packet switches supporting variable length packets. In Section 3, we describe packet scheduling algorithms that determine the wavelength of packets inserted in the FDL buffer, and then present our new algorithm. In Section 4, we introduce the simulation model and evaluate the two presented architectures. Finally, in Section 5, we discuss the feasibility of packet scheduling algorithms. Conclusions and future studies are

summarized in Section 6.

2 Photonic Packet Switch Architectures

Figure 1 shows the diagram of a photonic packet switch. As shown in this figure, the operation and implementation of photonic packet switches include four sub blocks: (a) a packet synchronization, (b) an input interface consisting of packet header information extraction and packet header removal, (c) an optical switch fabric consisting of space switch and FDL buffer, (d) an output interface which inserts a new header. The photonic packet switches that we consider in this thesis accept variable-length packets arriving asynchronously at the input port. Arriving packets are synchronized at a time with a predefined size. A synchronization mechanism for asynchronously arriving packets is presented in [15], see also Fig. 2.

The packet length is an integer multiple of the time slot size. When we utilize FDL, the time slot size affects the performance of the switch when the variable-length packets are treated. For example, in [19], it is shown that the best performance is obtained when the time slot size is set to about 30 percent of the average packet size. We will also use this value in the simulation experiments presented in Section 4.

The photonic packet switch is equipped with wavelength converters and optical buffers in order to resolve contentions of packets. A number W of the wavelengths are multiplexed on the fiber and the packets are carried on the wavelength. The wavelengths are demultiplexed at the input port of the switch. The packet on the wavelength is then time-synchronized at the time slot. Then, the packet scheduling algorithm determines the destination of each arriving packet. If the corresponding output port is available, the packet is sent to the output port directly after being assigned the appropriate wavelength. Otherwise, it is inserted in the optical buffer according to the scheduling algorithm. The scheduled packets are sent through a space switch. The wavelength of the packet is converted to the proper wavelength by a fixed wavelength converter at the output port.

One FDL buffer is consists of a number B of delay lines, which are set up in parallel. The

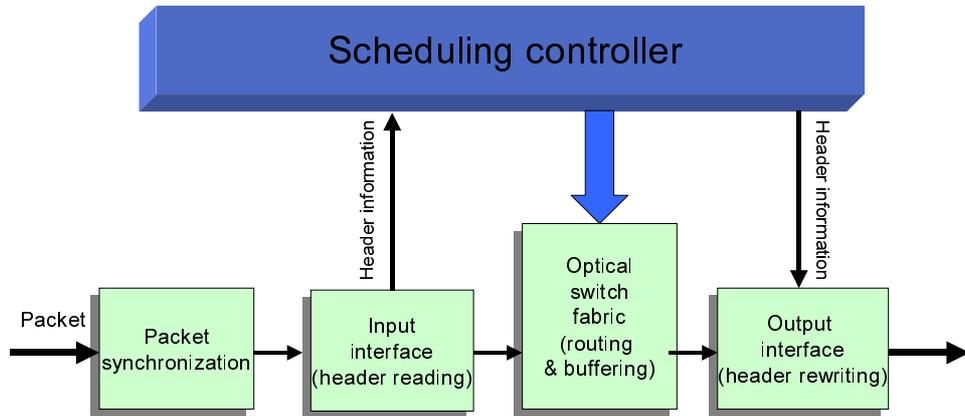


Figure 1: Block diagram of photonic packet switch

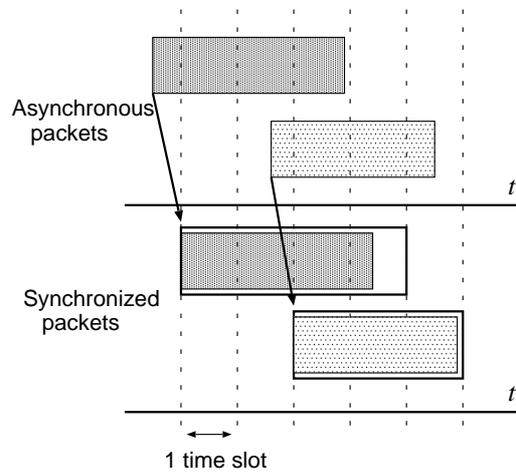


Figure 2: Synchronization of packets inside a switch

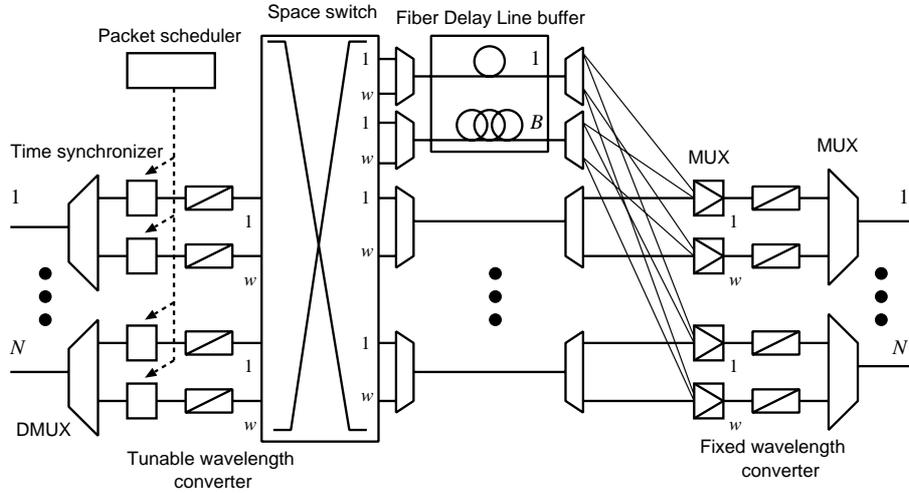


Figure 4: Shared buffer type photonic packet switch architecture

Figure 4 shows the architecture of the shared buffer type switch, which has one shared FDL buffer, and the packets are stored at the same buffer regardless of the destination output port. As in the output buffer type switch, when the contention cannot be resolved by wavelength conversion, the packets are sent to the FDL buffer. When the contention of packets can be resolved by wavelength conversion, on the other hand, the packets are sent to the output ports directly. The shared buffer type switch has only one FDL buffer with W virtual input lines. The buffer size B_T is equal to B .

The ratio of the number of switch inputs to buffer inputs is $N : 1$, thus the switch performance is likely to be degraded. One possible way to resolve this problem is to increase the number of wavelengths multiplexed on FDL (W_i), by which more packets can be stored in parallel at one time. However, W_i wavelengths should be decreased to W (the number of wavelengths on the output port line), and therefore, careful packet scheduling becomes necessary. That is, in order to prevent the contentions of the packets in output ports, the scheduling algorithm needs to determine the internal wavelength and the external wavelength for every packet. Furthermore, we

need additional wavelength converters for that purpose. In Section 4, we will evaluate the effect of this technique by conducting simulation experiments. It should be noted here that this method is only applicable to the shared buffer type switch. In the output buffer type switch, it does not help improving the performance since each output port is equipped with one FDL buffer.

3 Packet Scheduling Algorithms

A packet scheduling algorithm is needed in order to determine the wavelength and FDL for the arriving packets. We assume that time is synchronized and multiple packets may arrive within the time slot. For each of the packets arriving within the time slot, the packet scheduler finds the appropriate wavelength and delay line as follows. If an unused wavelength on the output port is found, the packet is sent to the output port directly. When no wavelength is available at the output port, the appropriate FDL is found.

3.1 Buffer Control Algorithms

In the following, we briefly introduce four algorithms (A0 through A3), followed by our enhancement which is applied to Algorithms A1, A2, and A3.

Algorithm A0: Assign the Wavelength in Round-Robin Fashion

One of simplest forms of algorithm is to assign the wavelength for packets arriving within the time slot in a round-robin fashion. This is simple and easy to implement. The information that the algorithm should hold includes (1) the latest number of the wavelength to which the previous packet is assigned, and (2) the queue lengths of the wavelengths. The latter can be implemented by using a counter associated with the wavelength, which is increased incrementally by the packet length (in time slot) when the wavelength is chosen by the algorithm and decreased decrementally by one at every time slot.

Algorithm A1: Assign to the Buffer with Minimum Queue [16, 17, 18]

Algorithm A1 assigns the packet to the wavelength with the minimum queue length. The order selection of the packet from among the ones arriving within the time slot is random, or is simply decided according to the input port number at which the packet has arrived. For this purpose, a

simple counter associated with the wavelength is utilized, as in Algorithm A0. Then, the appropriate FDL is selected for the packet to be sent to. If the FDL buffer is full, the packet is discarded. This algorithm is simple and packet scheduling is easy to implement because the procedure used by the scheduler only seeks the minimum queue length for each packet.

Algorithm A2: Assign the Shortest Packet First to Wavelength with Minimum Queue [16]

Algorithm A2 first sorts packets arriving within the time slot into an order of increasing packet length. It then assigns the wavelength with the minimum queue length to the shortest packet. Then, it updates the queue counter for the chosen wavelength and finds the wavelength with the minimum queue length for the second shortest packet. This process is iterated until the destinations of all the packets are determined. This algorithm needs to perform sorting of input packets and to find the wavelength with minimum queue length for each packet. Since the maximum number of packets arriving within the time slot is $N \times W$, it is complicated and the scheduler needs to have a high processing speed.

Algorithm A3: Assign the Longest Packet First to Wavelength with Minimum Queue

In contrast to Algorithm A2, Algorithm A3 sorts wavelengths for the packets into an order of decreasing packet length. Then, the same procedure is performed as in Algorithm A2. Computational complexity is the same as for Algorithm A2. By using Algorithm A3, more information is carried at the expense of losing shorter packets and increasing the packet loss probability.

3.2 Void Space Reduction Method

In order to prevent errors in the ordering of packets, a switch processes packets in order of arrival. Thus, when a packet is sent to FDL, a newly arriving packet with same input/output ports as those of the previously arriving packet should not be sent to the shorter FDL. The previously reported algorithms, except for Algorithm A0, have this feature. However, this feature results in

unacceptable performance degradation as discussed in the next section.

Our void space reduction method proposed in this subsection is applicable to the shared buffer type switch. Since the output buffer type switch is equipped with the FDL buffer for each output port, the buffer packet is sent to the destination output port using the wavelength assigned to the FDL. On the other hand, the number of output ports of the shared buffer type switch is larger than the total of buffer inputs. Therefore, the use of the packet scheduling method, in which the same wavelength is used for the FDL and output port, leads to less utilization of output ports and an overload at the FDL buffer. Thus, the void space reduction method presented below is useful for the shared buffer type switch.

Since the shared buffer type switch has a single buffer, the queue length of the buffer increases under high traffic load conditions. Consequently, the output interval between two packets destined for the same output port becomes large, which is called *void space* in this thesis. As an example, Fig. 5 illustrates why and how void space appears. At output port 1, a packet is sent on wavelength w_1 . The queue counter is then increased by the packets sent to output ports 2 and 3. Now, a new packet destined for output port 1 arrives at the switch. If the packet is assigned wavelength w_1 , the packet will be stored at the back of the queue of the buffer because wavelength w_1 of output port 1 is in use. Then, a void space of length 4 appears, leading to low utilization of output port 1. In this case, it is impossible to use output port 1 until all the buffered packets are transmitted, regardless of whether or not the port is actually in use.

Incidentally, in the strict sense of the word, void space and excess load are used in [20] and [21], respectively. However their definitions are different from those of void space and excess load use in this thesis. Void space and excess load are actually empty spaces between asynchronously arriving packets that are destined for the same output port. The void space considered in this thesis is the portion of fiber in which the packets for different output ports are stored between two packets for the same output port, this portion causes the low utilization of output ports. In [20], a void filling algorithm has been proposed. However, when using this algorithm, the

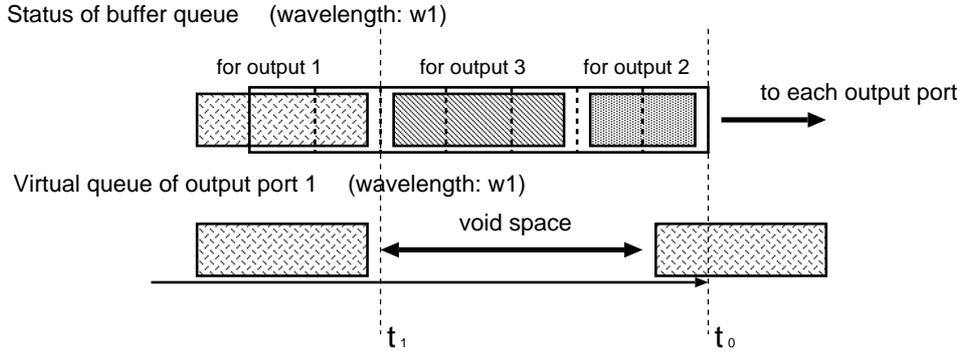


Figure 5: Void space in shared buffer type switch

packet scheduler should maintain the arriving/departing times of all packets stored in the buffer in order to insert a new packet within void space. Therefore, this algorithm is highly complex and is difficult to implement.

Our proposal, called the *void space reduction method*, reduces the adverse effect of void space by wavelength conversion. The wavelength of the packet is converted so that the influence of void space is minimized. Figure 6 illustrates our approach. Suppose that a new packet destined for output port 1 arrives at a switch. The packet is assigned wavelength w_1 and is stored in the buffer. If the next arriving packet is assigned wavelength w_1 , void space between two time slots appears. On the other hand, our method compares the queue lengths of the wavelength buffers and selects a wavelength which will minimize void space. In the above case, therefore, the new packet is assigned wavelength w_2 , and there by preventing formation of void space completely. Note that this method can be applied to Algorithms A1 through A3.

More specifically, our method works as follows. To implement our method, we introduce a *virtual queue* within the physical shared buffer. A virtual queue is a logical queue maintained for each combination of the output port and wavelength on the output fiber. Thus, there are a number $N \times W$ of virtual queues in the shared buffer. We also introduce a counter for maintaining the output time of the last packet in the virtual queue. When a new packet arrives and is decided to be

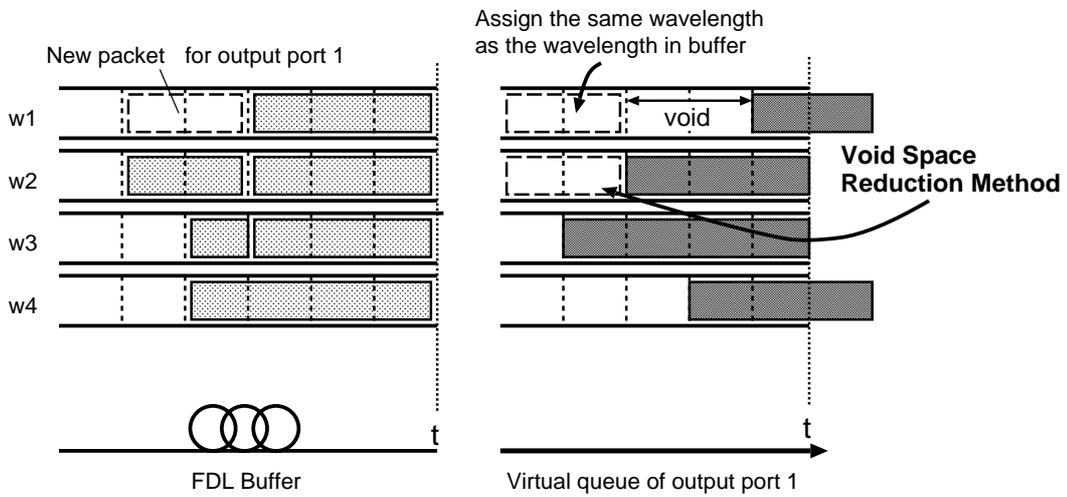


Figure 6: Void space reduction method

stored in the buffer (i.e., because no available wavelength is found), the scheduler finds the smallest difference between the physical queue length of the wavelength and the virtual queue counter. Then, the packet is inserted into FDL. After the packet goes through the FDL, the wavelength of the packet is tuned to the wavelength that is actually used for the output fiber.

Lastly, it should be noted that in order to implement this method, wavelength conversion is necessary, which leads to a higher switch cost, but the improvement in performance is remarkable, as discussed in the next section.

4 Performance of the Photonic Packet Switches

4.1 Simulation Model

For comparative evaluation, the photonic packet switch and arriving traffic are modeled as follows. The numbers of input/output ports N and wavelengths on the fiber W are set to be 16 and 8, respectively. The wavelength capacity is 40 Gbps. A packet arrives according to a Poisson process. The average packet length is 400Bytes. The packet length is exponentially distributed, but truncated at 1000Bytes. The time slot size is 20ns, which corresponds to 30% of the average packet length [19]. Every input fiber and wavelength has the same packet arrival rate, and the destination output port of the packet is chosen randomly.

4.2 Evaluation of the Packet Scheduling Algorithms

In this subsection, we evaluate the packet scheduling algorithms A0 through A3 described in Section 3. Figures 7 and 8 show the simulation results of packet loss probability dependent on the buffer size B_T (the total number of delay lines in the whole switch) in the output buffer type switch and the shared buffer type switch, respectively. Figures 9 and 10 show the simulation results in the output buffer type switch and the shared buffer type switch, respectively when the load is 0.6. Figures 11 and 12 show the simulation results in the output buffer type switch and the shared buffer type switch, respectively when the load is 0.8. As shown in Figs. 7, 9, 11, algorithms A1 through A3 give better performance than algorithm A0 under any traffic load condition, and algorithm A2 gives the best performance. The packet loss probabilities of the shared buffer type switch differ greatly from those of the output buffer type switch especially under the high traffic load condition, as shown in Figs. 8, 10, 12. In the low traffic load condition, algorithm A2 again gives the best performance.

In Figs. 8, 10, 12, the performance of the shared buffer type switch decreases when the switch is equipped with a larger buffer size. This is because the queue length becomes long and the

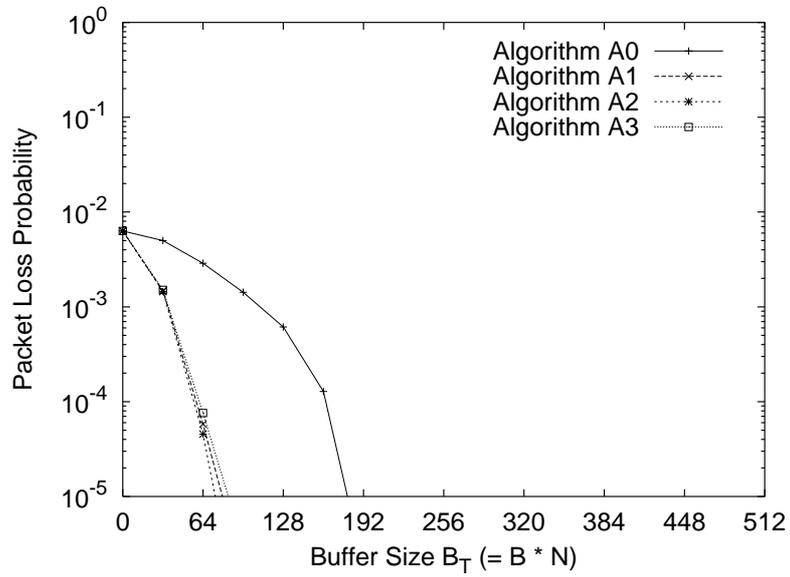


Figure 7: Packet loss probability (output buffer type switch, load = 0.4)

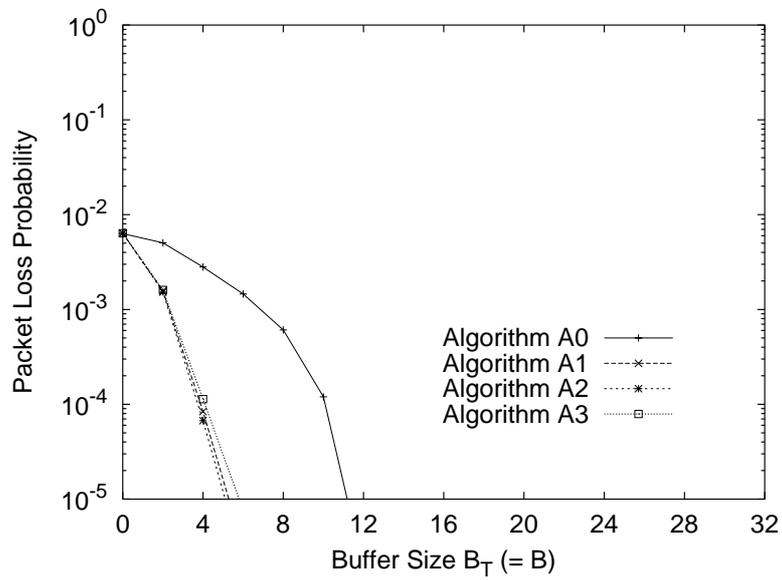


Figure 8: Packet loss probability (shared buffer type switch, load = 0.4)

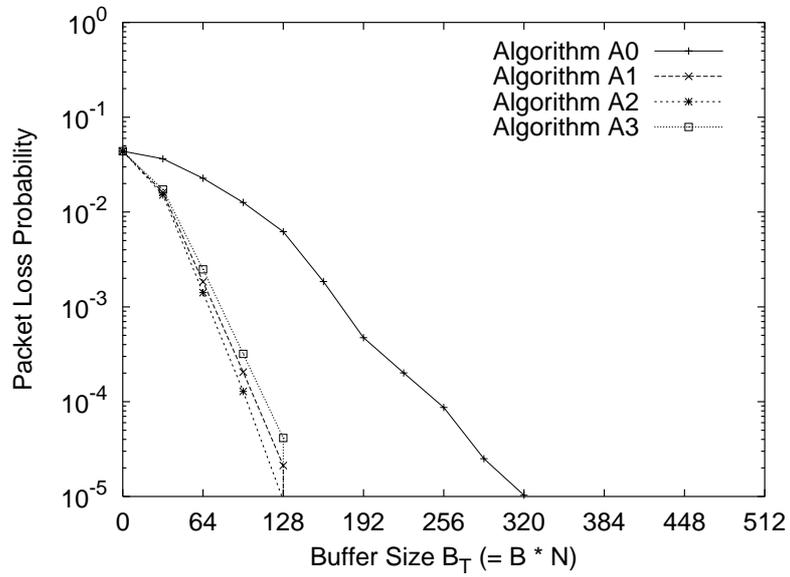


Figure 9: Packet loss probability (output buffer type switch, load = 0.6)

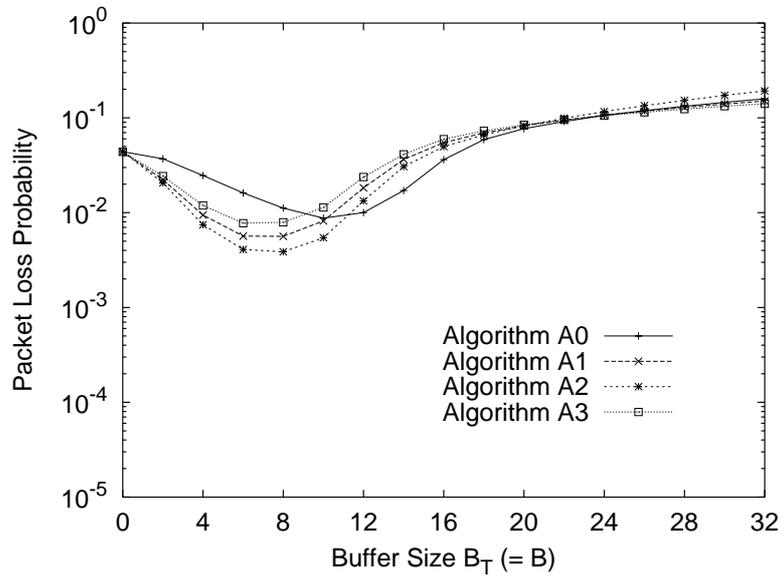


Figure 10: Packet loss probability (shared buffer type switch, load = 0.6)

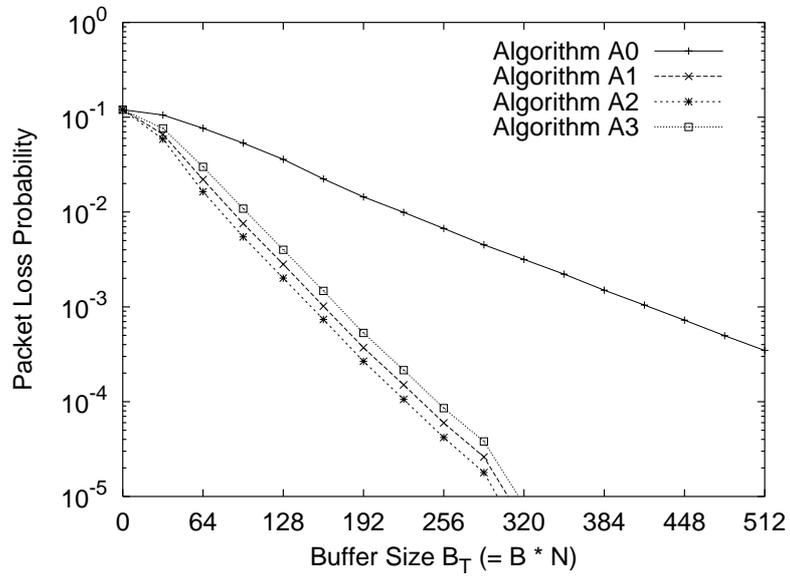


Figure 11: Packet loss probability (output buffer type switch, load = 0.8)

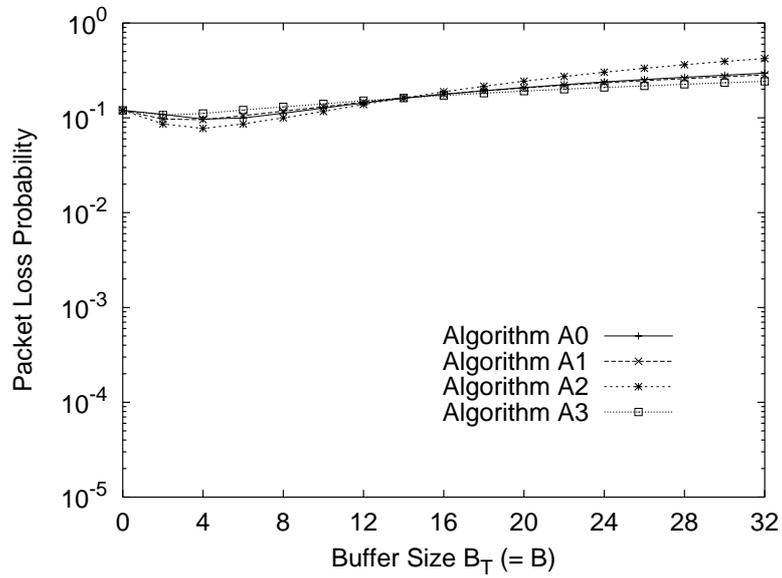


Figure 12: Packet loss probability (shared buffer type switch, load = 0.8)

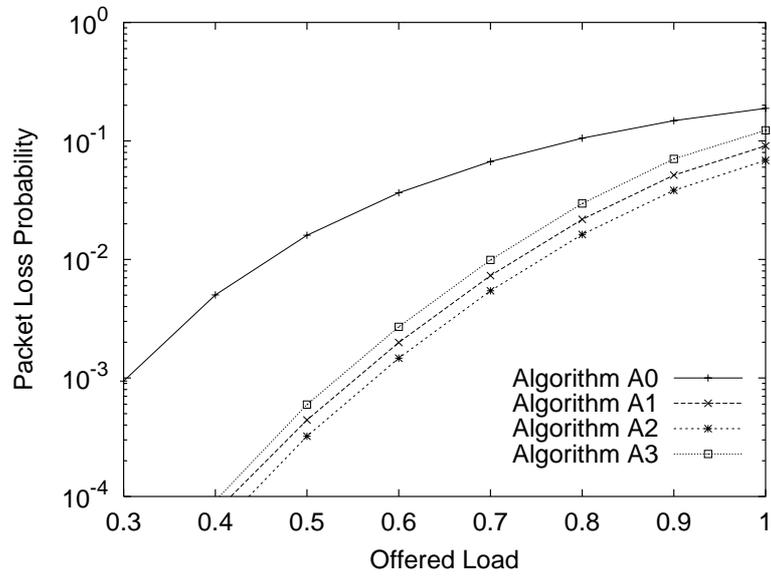


Figure 13: Packet loss probability (output buffer type switch, for $B = 64$)

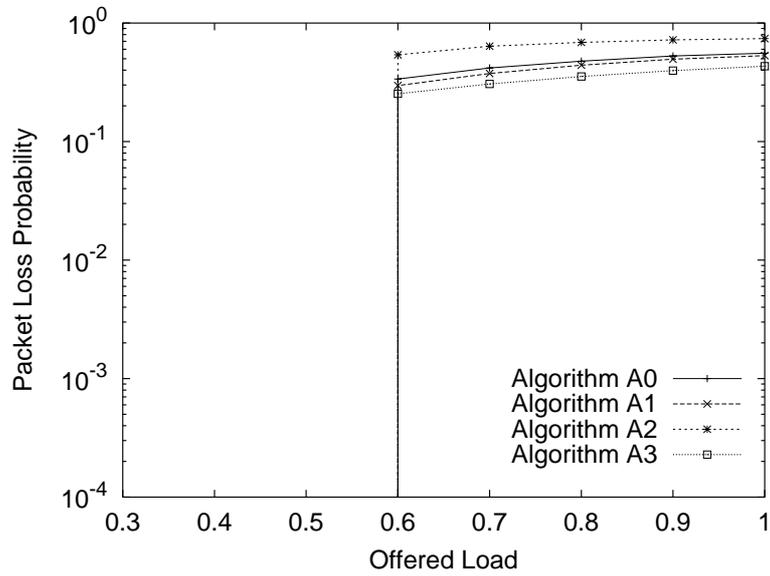


Figure 14: Packet loss probability (shared buffer type switch, for $B = 64$)

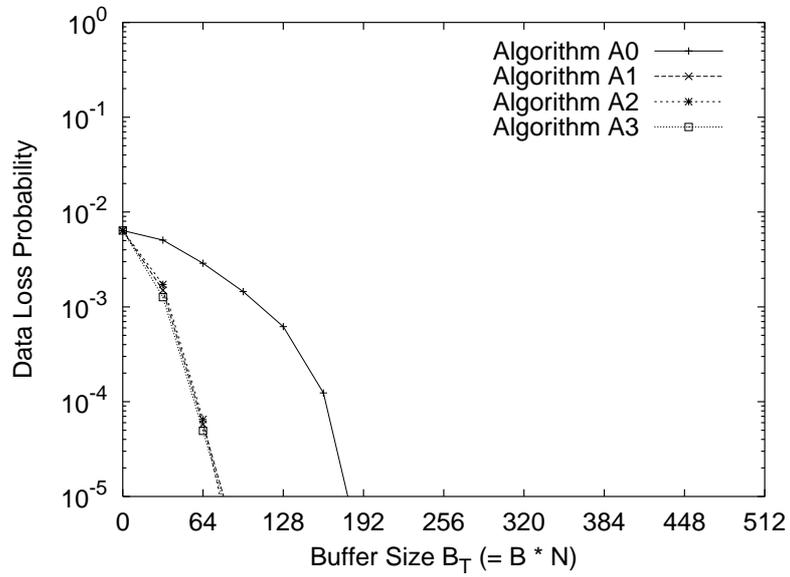


Figure 15: Data loss probability (output buffer type switch, load = 0.4)

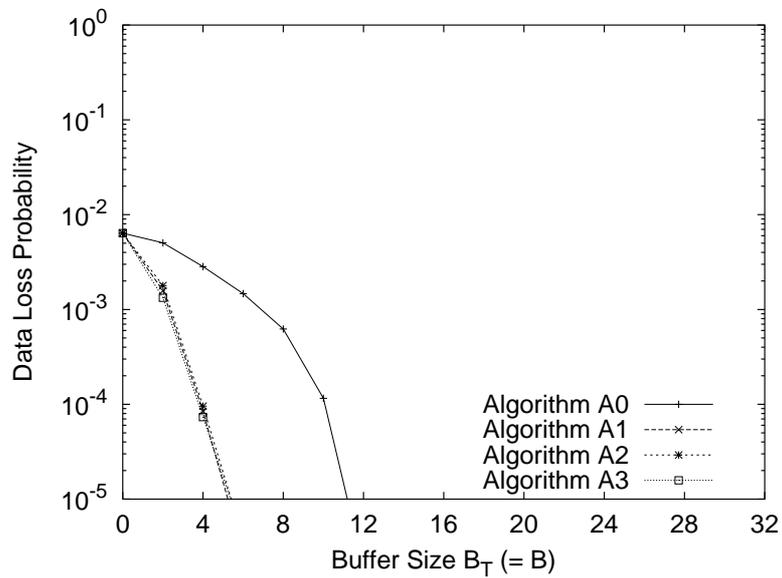


Figure 16: Data loss probability (shared buffer type switch, load = 0.4)

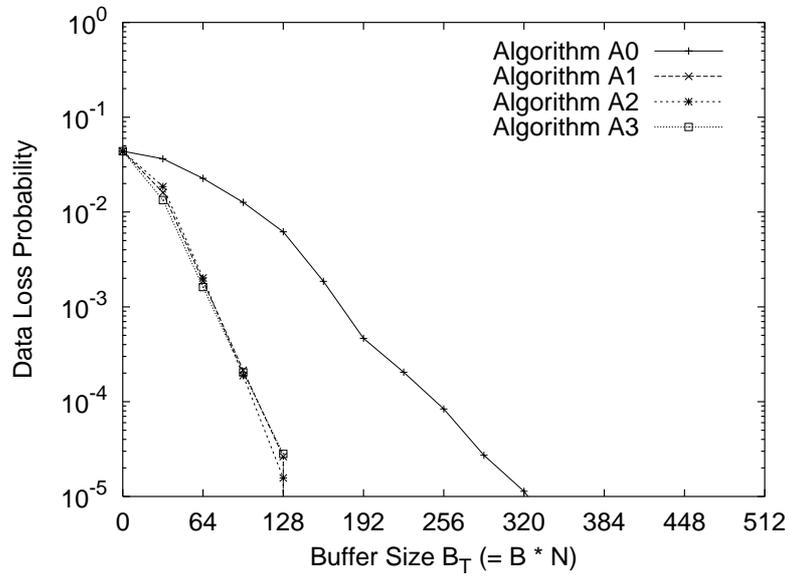


Figure 17: Data loss probability (output buffer type switch, load = 0.6)

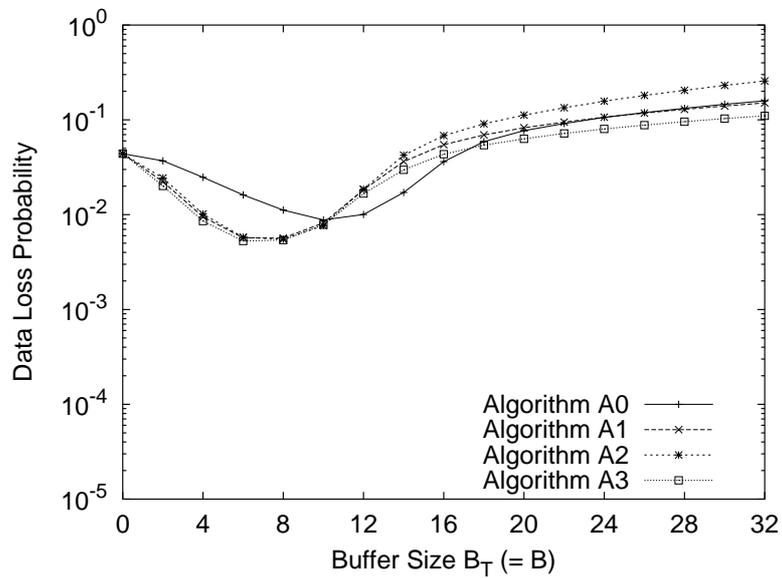


Figure 18: Data loss probability (shared buffer type switch, load = 0.6)

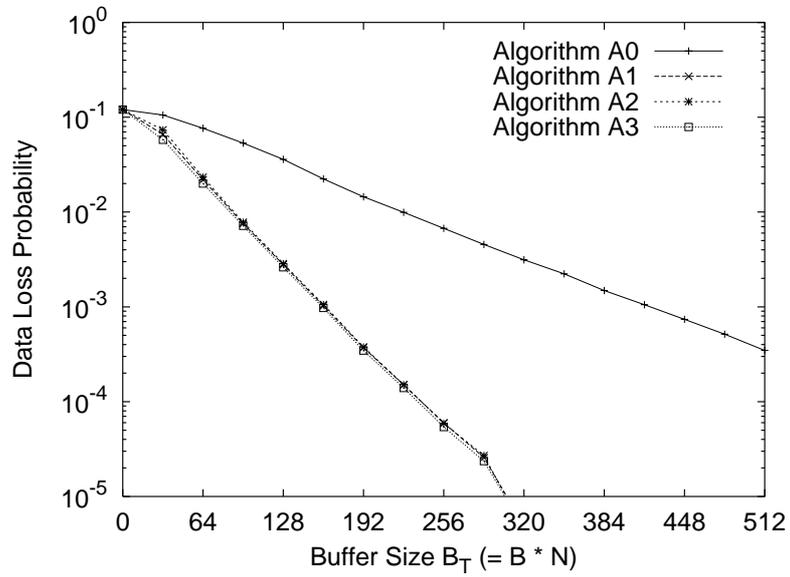


Figure 19: Data loss probability (output buffer type switch, load = 0.8)

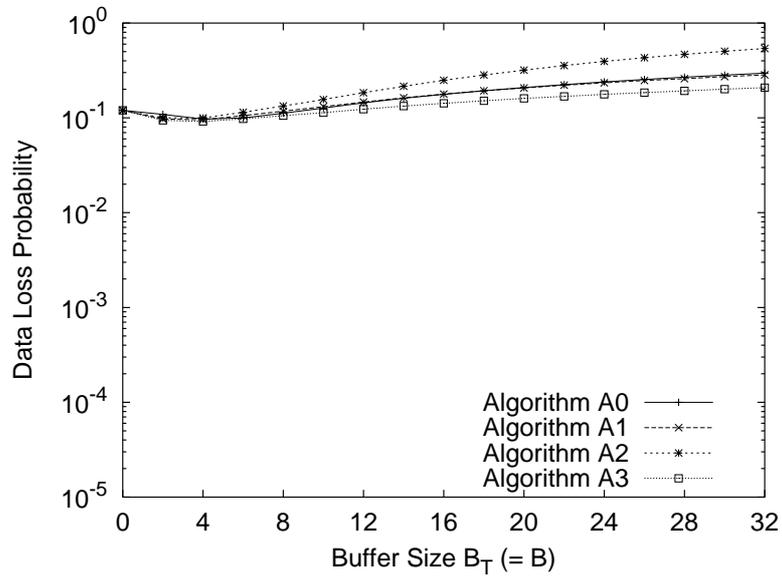


Figure 20: Data loss probability (shared buffer type switch, load = 0.8)

possibility of a void space appearing becomes high, as was described in Section 3.3. Figures 13 and 14 show the simulation results for the output buffer type switch and the shared buffer type switch, respectively, when the buffer size B_T is fixed at 64. In Fig. 13, it can be observed that the packet loss probability is gradually increased by the higher traffic load. On the other hand, the performance of the shared buffer type switch suddenly deteriorates as shown in Fig. 14. This is because input packets are continuously dropped as the buffer queue length becomes long under the high traffic load condition. The shared buffer type has an advantage in that it requires a smaller buffer size, in the current case, for a number of FDL. However, the performance of the shared buffer type switch deteriorates much more than that of the output buffer type switch under high traffic conditions. In the next subsection, we will demonstrate how our void space reduction method improves the performance of the shared buffer type switch.

Figures 15 and 16 plot data loss probabilities dependent on the buffer size B_T (the total number of delay lines in the whole switch) in the output buffer type and the shared buffer type switch, respectively when the load is 0.4. Figures 17 and 18 show the simulation results in the output buffer type switch and the shared buffer type switch, respectively when the load is 0.6. Figures 19 and 20 show the simulation results in the output buffer type switch and the shared buffer type switch, respectively when the load is 0.8. The data loss probability is defined as the ratio of the total amount of dropped packets to the total amount of input packets. The set of six figures (Figs. 15, 16, 17, 18, 19 and 20) shows the same tendency as the previous set of figures for packet loss probability (Figs. 7, 8, 9, 10, 11 and 12), but algorithm A3 achieves the best result for data loss probability because it gives preference to long packets when assigning the wavelength, thus more data is carried.

4.3 Evaluation of Void Space Reduction Method

In this subsection, we evaluate our proposed void space reduction method. Figure 21 shows the performance of the shared buffer type switch when the void space reduction method is applied.

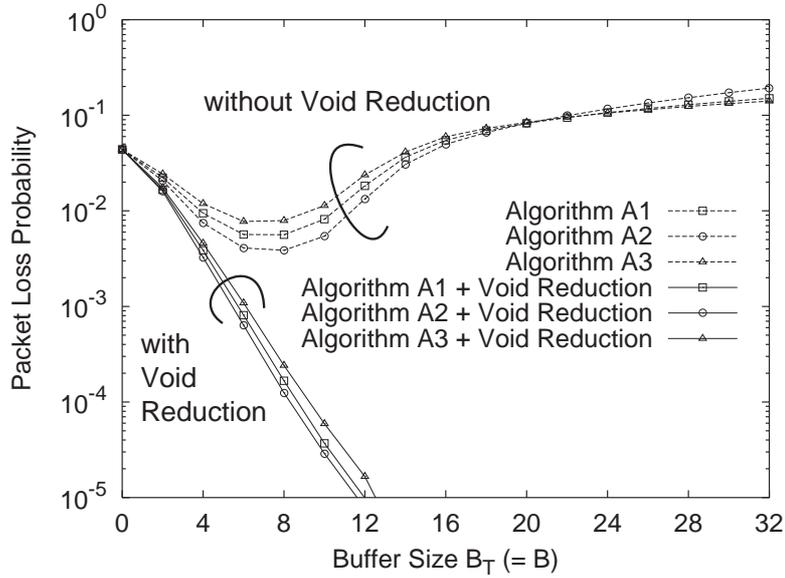


Figure 21: Packet loss probability (shared buffer type switch, void space reduction method, load = 0.6)

From this figure, it can be observed that the performance is dramatically improved by introducing the void space reduction method. And the shared buffer switch outperforms the output buffer switch by using the void space reduction method even with the smaller buffer.

4.4 Effects of Increasing the Number of Wavelengths on FDLs

Lastly, we last show the effects of increasing the number of wavelengths on FDLs (W_i). In Fig. 22, we plot the packet loss probability of the shared buffer type switch when the wavelengths on FDLs are increased ($W_i = 8, 16, 24$). From this figure, it can be observed that when the switch can store more packets in the buffer at one time the performance is actually improved. Of course, the void space reduction method can further improve this performance, and this is demonstrated in Fig. 23.

From these two figures, it is clear that the performance of the shared buffer type switch when using the void space reduction method is even better than that of the output buffer type switch.

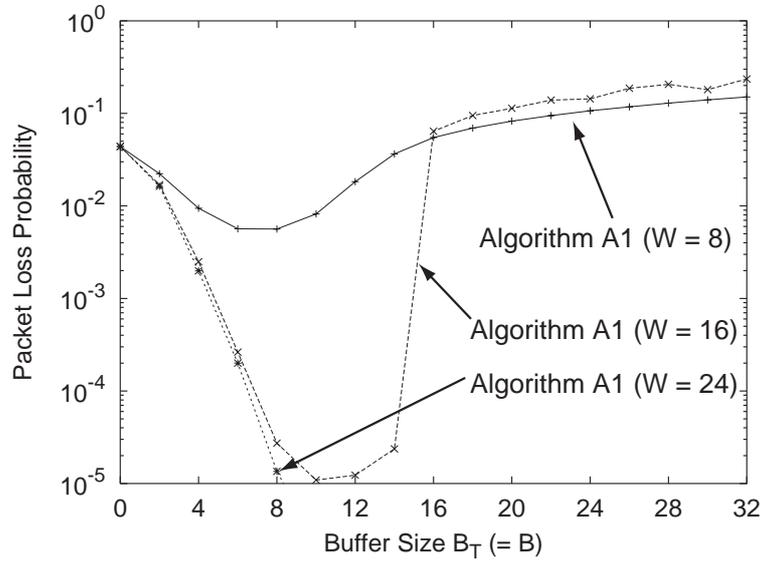


Figure 22: Packet loss probability (shared buffer type switch, increasing the inner wavelengths, load = 0.6)

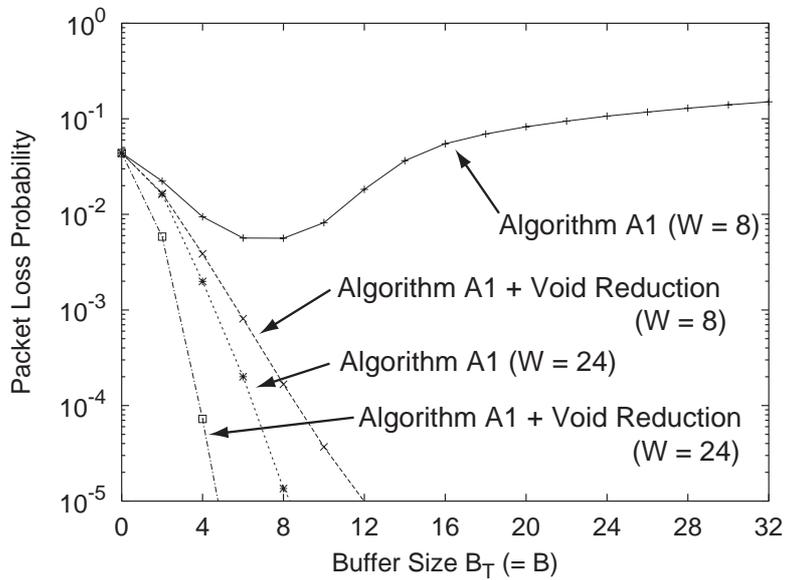


Figure 23: Packet loss probability (shared buffer type switch, void space reduction method + increasing the inner wavelengths, load = 0.6)

5 Feasibility of Packet Scheduling Algorithms

In the simulation results, we have observed that our proposed method can achieve the drastic improvement in the performance. However, there is another issue that should be discussed, that is, whether these algorithms are actually executable, or how much progress of hardware devices is required in order to implement these algorithms. In this section, we discuss this issue from the viewpoint of processing delay time using the algorithms.

5.1 PLD Implementation and Simulation

In order to consider the feasibility of packet scheduling algorithms, we design the hardware architecture for the algorithm using PLD design software. We describe the source code in VHDL (VHSIC Hardware Description Language), and we measure the processing delay time of the algorithms through simulations.

Hereafter, we explain the details of the PLD implementation and simulation. The PLD design tool that we used is MAX+PLUS®II software by Altera Corporation [22]. First, we consider the scheduling phase of the algorithm by dividing it into three separate parts. The reason for introducing the scheduling phase is that we want to simplify the implementation of the algorithms.

We define three scheduling phases as follows:

- Phase P1: Zero decision of the queue length, which examines whether free wavelength exists in the output port. If the queue of which length is zero exists, the packet is sent to the destination output port directly. Otherwise, it is inserted into the optical buffer. This scheduling phase is necessary in algorithms A0 and A1.
- Phase P2: Comparison of buffer queue and virtual queue, which compares the length of the buffer queue with that of the virtual queue in each wavelength. This is necessary in algorithms A0 and A1.

- Phase P3: Search for the Shortest Queue, which searches the queues in the buffer or the destination output port for the shortest one. This is necessary in algorithm A1 and the void space reduction method.

In the packet scheduling algorithm, other processes including increasing the number of counters are required. However, we ignore such processing time since their processing delay time are much smaller than those of the above three scheduling phases.

We now take a close look at the simulation results. Figure 24 shows that processing delay time depends on the number of wavelengths in the fiber for each scheduling phase. As shown in the figure, the processing delay time in phase P2 is fixed regardless of the number of wavelengths. This is because the comparison of two queue lengths can be processed simultaneously. On the other hand, those in phases P1 and P3 increase as the number of wavelengths increases, because the procedures of phases P1 and P3 are sequential depending on the number of wavelengths. In Fig. 25, we derive the processing delay time for packet scheduling algorithms based on the results in Fig. 24. In this figure, the processing time of scheduling phase P3 is a burden for packet scheduling algorithms. For example, when the number of wavelengths is 8, the processing delay time of algorithms A0, A1 and A1 with void space reduction are 8.6, 23.9 and 39.2ns, respectively. Therefore, it can be concluded that algorithms which include the procedure of phase P3 requires a much longer processing delay time for their execution. However, we can reduce the processing delay time for the packet scheduling algorithms with the reduction of relevant wavelengths, as described in the next subsection.

5.2 Wavelength Grouping Method

If a too complicated packet scheduling algorithm requiring slow electronic is applied to a switch, it is difficult for the controller to process all the arriving packets in one slot time. Therefore, it is important to introduce the packet scheduling algorithm that poses a reasonable processing delay time within a time slot. One effective way of reducing the processing delay time is to decrease

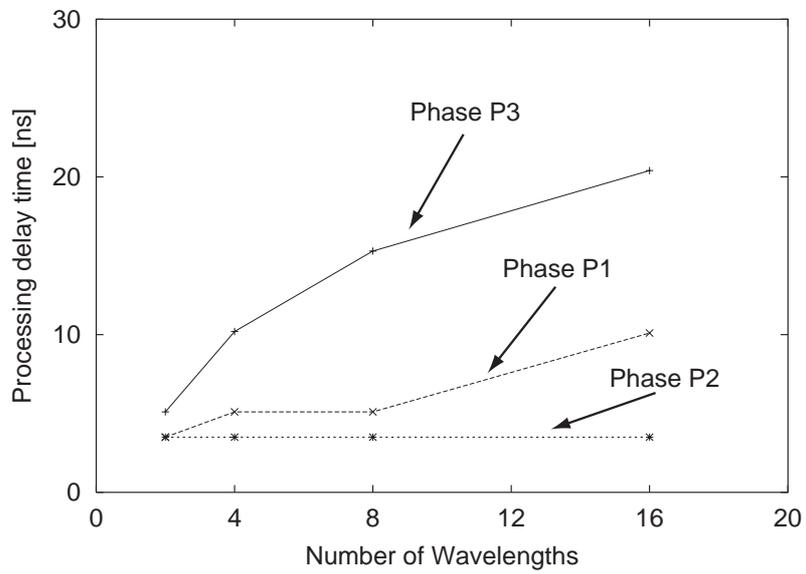


Figure 24: Processing delay times for scheduling phases

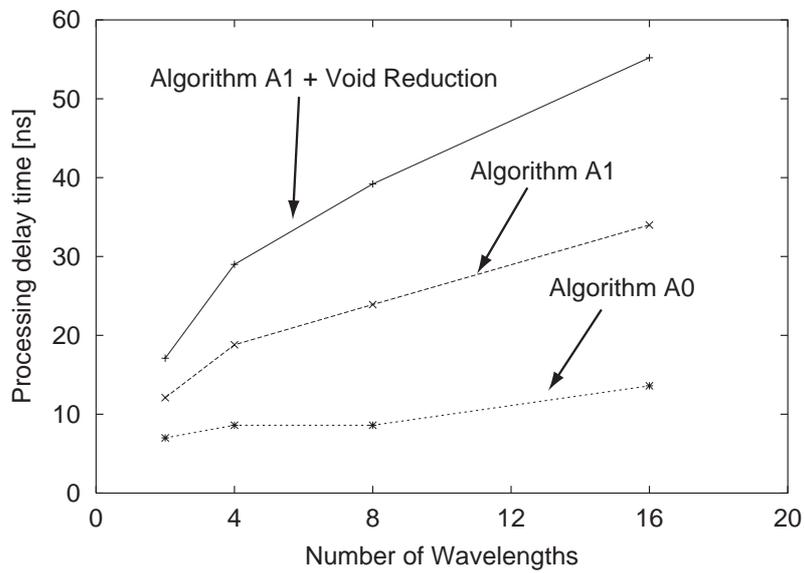


Figure 25: Processing delay times for packet scheduling algorithms

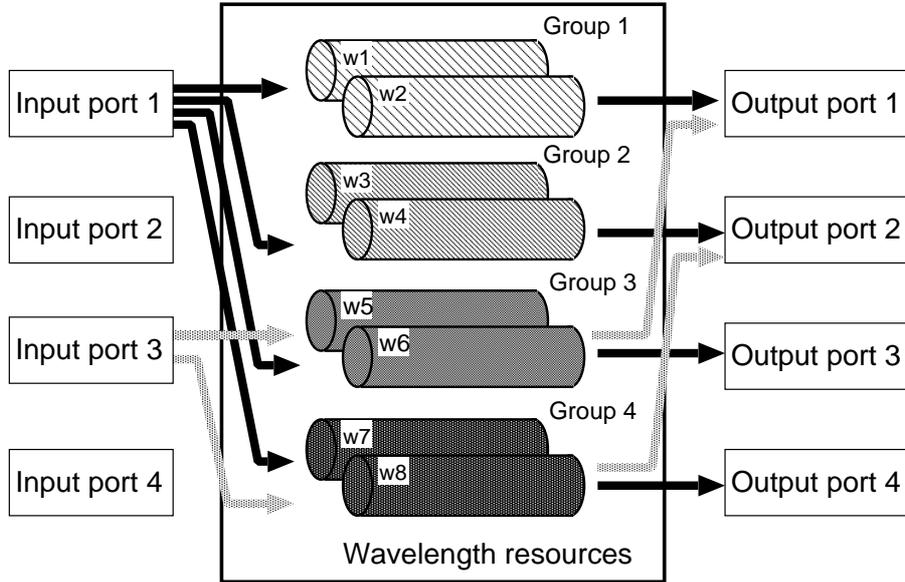


Figure 26: An example of wavelength grouping method (4 × 4 switch, 8 wavelengths, 4 groups)

the number of wavelengths that the scheduler has to examine, because the processing delay time largely depends on the number of wavelengths, as discussed in the previous section. To realize this, we propose the *wavelength grouping method*, which divides the wavelengths on a fiber into several groups and determines the group depending on the combination of its input port and output port for each packet. The wavelength in the selected group is assigned to a packet. We illustrate an example in Fig. 26 and Tab. 1. When N numbers of input/output ports and W wavelengths on a fiber are four and eight (from w_1 to w_8), respectively, and we set four groups of wavelengths, the packet that arrives at input port 1 and is destined for output port 1 will be assigned wavelength w_1 or wavelength w_2 , and the packet that arrives at input port 2 and is destined for output port 3 will be assigned wavelength w_3 or w_4 . In this way, the processing time for wavelength search can be reduced. Of course, the range of wavelength selection becomes narrow. Figures 27 and 28 show the simulation results of packet loss probability in the switch to which the wavelength grouping method is applied. As shown in these figures, as the number of groups increases, the performance

Table 1: Wavelength groups (4×4 switch, 8 wavelengths, 4 groups)

| input \ output | 1 | 2 | 3 | 4 |
|----------------|------------|------------|------------|------------|
| 1 | w_1, w_2 | w_3, w_4 | w_5, w_6 | w_7, w_8 |
| 2 | w_7, w_8 | w_1, w_2 | w_3, w_4 | w_5, w_6 |
| 3 | w_5, w_6 | w_7, w_8 | w_1, w_2 | w_3, w_4 |
| 4 | w_3, w_4 | w_5, w_6 | w_7, w_8 | w_1, w_2 |

Table 2: Technology road map for performance of ASIC [1]

| Year | 2000 | 2002 | 2004 | 2008 | 2014 |
|-----------------|------|------|------|-------|-------|
| Frequency (MHz) | 559 | 700 | 828 | 1,200 | 1,800 |

of a switch is degraded. For example, when the number of groups is more than 4, even the void space reduction method cannot achieve superior performance. Therefore, it is important to consider the trade-off between the performance of the switch and the processing delay time of packet scheduling algorithms.

5.3 Consideration of feasibility of packet scheduling algorithms

The simulation results for the implementation of packet scheduling algorithms have shown that it is difficult to apply as a switch with the present technology of electronic devices. However, the progress in semiconductor technology is remarkable, which predicts the feasibility of packet scheduling algorithms in the near future.

For example, Tab. 2 shows the technology road map for ASIC (Application-Specific Integrated Circuit) [1]. The road map predicts that the chip frequency will improve twofold or more within 8 years. If this prediction turns out to be correct, the processing delay time of algorithm A1 with the

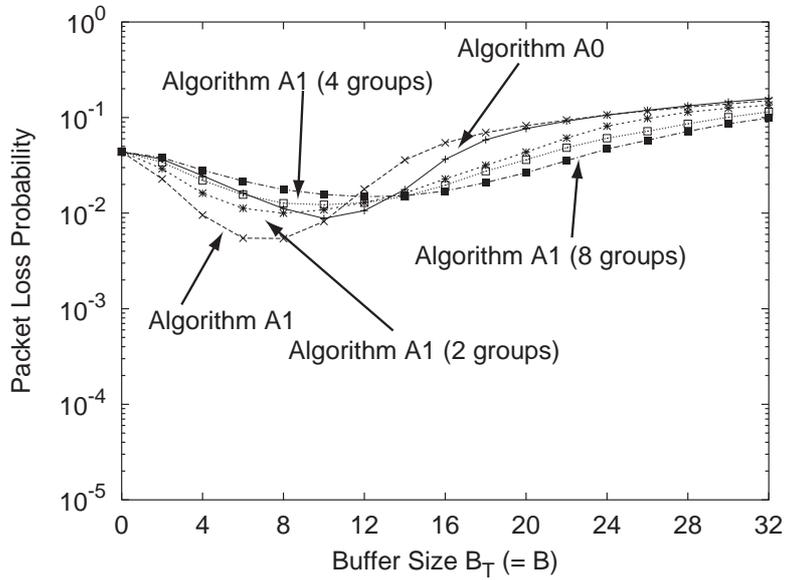


Figure 27: Packet loss probability (wavelength grouping method, algorithm A1 without void space reduction method, load = 0.6)

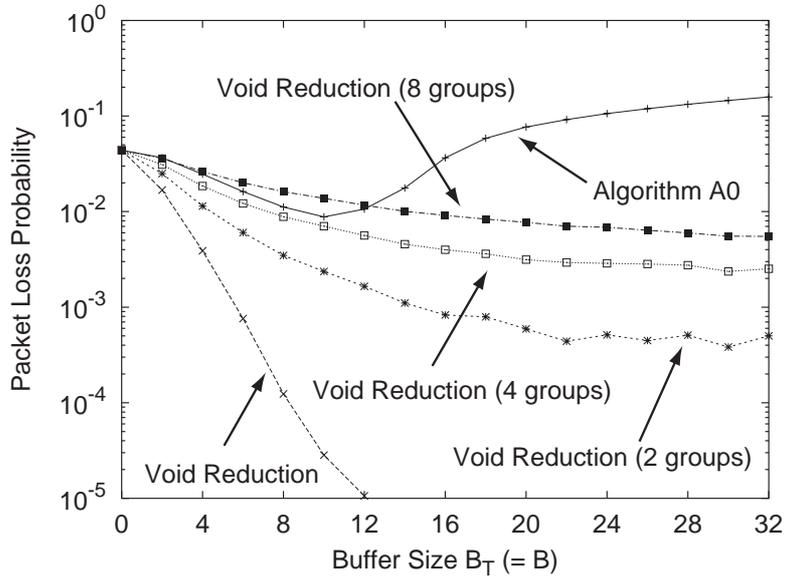


Figure 28: Packet loss probability (wavelength grouping method, algorithm A1 with void space reduction method, load = 0.6)

void space reduction method will be decreased to less than 20ns about 10 years after. It encourages us to expect the realization of packet scheduling algorithms. Until we get such a technology, we can make the switch to work within processable time by using the wavelength grouping method we proposed. As shown in the simulation results, for example, when the number of the groups is 4 (the number of processed wavelengths per a packet is 2), the processing delay time can be decreased below one slot time that we used in the simulations. Therefore, using our proposed algorithm, we can achieve the maximum performance of the switch when we decide the proper number of wavelengths according to the limitation of the hardware processing ability.

6 Conclusion

In this thesis, we evaluated the performance of the shared buffer type switch and output buffer type switches by applying packet scheduling algorithms. We compared the architectures of these two switches taking into account the total number of FDLs. Our simulation results showed that the shared buffer type switch achieves better performance than the output buffer type switch under low traffic load conditions. On the other hand, under high traffic load conditions, the output buffer type switch gives much better performance than the shared buffer type switch. However, our void space reduction method can improve the performance of the shared buffer type switch even better than that of the output buffer type switch. Furthermore, we discussed the feasibility of packet scheduling algorithms from the viewpoint of processing delay time. Our wavelength grouping method decreased the processing delay time which depends on the number of wavelengths. Therefore, using our proposed algorithm, we can achieve the maximum performance of the switch when we decide the proper number of wavelengths according to the limitation of the hardware processing ability. Moreover, we considered the feasibility based on hardware implementation and a technology road map.

In future studies, we need to consider a more effective packet scheduling algorithm from the viewpoint of the reduction of scheduling time and improvement in the performance of the switch.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Professor Masayuki Murata at Osaka University, who introduced me to the area of computer networks including the subjects in this thesis and support through my studies. His excellent guidance and thoughtful advice have provided an invaluable experience that will help me in my career.

I would like to express my sincere appreciation to Professor Ken-ichi Kitayama at Osaka University for his encouragement and invaluable comments through my studies and preparation this thesis.

All works of this thesis would not have been possible without the helpful suggestions and support of Associate Professor Ken-ichi Baba at Osaka University. He has opened the way for the accomplishment of this thesis.

My special thanks are due to Research Assistant Shin'ichi Arakawa at Osaka University, who gave me many advises.

I am deeply grateful to Associate Professor Yoshinori Takeuchi at Osaka University for his appreciated comments.

I am also indebted to Professor Hideo Miyahara, Associate Professor Naoki Wakamiya, Associate Professor Hiroyuki Ohsaki, Associate Professor Go Hasegawa, Assistant Professor Ichinoshin Maki at Osaka University who gave me helpful comments and feedbacks.

Furthermore, I thank all my colleagues in the Department of Informatics and Mathematical Science of Osaka University for their generous help, enlightening and valuable suggestions.

Finally, I wish to express my warmest thanks to my parents, grandparents and sister for their unconditional love, support, patience and understanding.

References

- [1] *International Technology Roadmap for Semiconductors 1999 Edition*. Semiconductor Industry Association, 1999.
- [2] R. S. Tucker and W. D. Zhong, "Photonic packet switching: An overview," *IEICE Transactions on Communications*, vol. E82-B, pp. 254–264, Feb. 1999.
- [3] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine*, pp. 136–142, Jan. 2001.
- [4] S. L. Danielsen, B. Mikkelsen, C. Joergesen, T. Durhuus, and K. E. Stubkjaer, "WDM packet switch architectures and analysis of the influence of tuneable wavelength converters on the performance," *IEEE Journal of Lightwave Technology*, vol. 15, pp. 219–227, Feb. 1997.
- [5] S. L. Danielsen, C. Joergesen, B. Mikkelsen, and K. E. Stubkjaer, "Analysis of a WDM packet switch with improved performance under bursty traffic conditions due to tuneable wavelength converters," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 729–735, May 1998.
- [6] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 2081–2094, Dec. 1998.
- [7] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic, "SLOB: A switch with large optical buffers for packet switching," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 1725–1736, Oct. 1998.
- [8] K. L. Hall and K. A. Rauschenbach, "All-optical buffering of 40-Gb/s data packets," *IEEE Photonic Technology Letters*, vol. 10, pp. 442–444, Mar. 1998.
- [9] S. Bregni, G. Guerra, and A. Pattavina, "Optical switching of IP traffic using input buffered architectures," *Optical Networks Magazine*, vol. 3, pp. 20–29, Nov. 2002.

- [10] S. L. Danielsen, B. Mikkelsen, C. Joergesen, T. Durhuus, and K. E. Stubkjaer, "Wavelength conversion in optical packet switching," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 2095–2108, Feb. 1997.
- [11] V. Eramo and M. Listanti, "Packet loss in a bufferless optical wdm switch employing shared tunable wavelength converters," *IEEE Journal of Lightwave Technology*, vol. 18, pp. 1818–1833, Dec. 2000.
- [12] K. E. Stubkjaer, A. Kloch, P. B. Hansen, H. N. Poulsen, D. Wolfson, K. S. Jepsen, A. T. Clausen, E. Limal, and A. Buxens, "Wavelength converter technology," *IEICE Transactions on Communications*, vol. E82-B, pp. 390–400, Feb. 1999.
- [13] 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," in *Proceedings of OptiComm 2000*, pp. 235–246, Nov. 2000.
- [14] S. Yao, S. J. B. Yoo, and B. Mukherjee, "A comparison study between slotted and unslotted all-optical packet-switched network with priority-based routing," in *Proceedings of OFC 2001*, Mar. 2001.
- [15] M. Murata and K. Kitayama, "Ultrafast photonic label switch for asynchronous packets of variable length," in *Proceedings of INFOCOM 2002*, vol. 1, pp. 371–380, June 2002.
- [16] A. Ge, L. Tancevski, G. Callegati, and L. Tamil, "WDM fiber delay line buffer control for optical packet switching," in *Proceedings of OptiComm 2000*, pp. 247–257, Nov. 2000.
- [17] F. Callegati, W. Cerroni, and G. Corazza, "Optimization of wavelength allocation in WDM optical buffers," *Optical Networks Magazine*, vol. 39, pp. 66–72, Nov. 2001.
- [18] L. S. Tamil, F. Masetti, T. McDermott, G. Castanon, A. Ge, and L. Tancevski, "Optical IP routers: Design and performance issues under self-similar traffic," *Journal of High-Speed*

Networks, vol. 8, pp. 59–67, June 1999.

- [19] F. Callegati, “Optical buffers for variable length packets,” *IEEE Communications Letters*, vol. 4, pp. 292–294, Sept. 2000.
- [20] L. Tancevski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti, and T. McDermott, “Optical routing of asynchronous, variable length packets,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2084–2093, Oct. 2000.
- [21] P. B. Hansen, S. L. Danielsen, and K. E. Stubkjaer, “Optical packet switching without packet alignment,” in *Proceedings of ECOC '98*, pp. 591–592, Sept. 1998.
- [22] Altera Corporation, “Max+plus II software.” available at <http://www.altera.com/products/software/pld/products/maxplus2/>.