フォトニックパケットスイッチにおける
パケット間空き領域の低減を考慮したスケジューリングアルゴリズム

山口 貴詩†　馬場 健一††　村田 正幸‡　北山 研一‡‡

† 大阪大学 大学院基礎工学研究科
〒 560–8531 豊中市待兼山町 1–3
Phone: 06–6850–6616, Fax: 06–6850–6589
E-mail: t-yamagt@ics.es.osaka-u.ac.jp

†† 大阪大学 サイバーメディアセンター
〒 567–0047 茨木市美穂ヶ丘 5–1
Phone: 06–6879–8793, Fax: 06–6879–8794
E-mail: baba@cmc.osaka-u.ac.jp

‡ 大阪大学 サイバーメディアセンター
〒 560–0043 豊中市待兼山町 1–3
Phone: 06–6850–6615, Fax: 06–6850–6589
E-mail: murata@cmc.osaka-u.ac.jp

‡‡ 大阪大学 大学院工学研究科
〒 567–0047 吹田市山田丘 2–1
Phone: 06–6879–7728, Fax: 06–6879–7688
E-mail: kitayama@comm.eng.osaka-u.ac.jp

あらまし　本稿では、WDM-FDL バッファを用いた可変長パケットを取り扱うことのできる 2 つのフォトニックパケットスイッチの性能比較評価を行う。すなわち、スイッチ内のすべての出力線にスイッチングされるパケットを 1 つの FDL バッファで共有して蓄積する共有バッファ型、および各出力ポートごとに設けられたそれぞれの FDL バッファに蓄積する出力バッファ型のフォトニックパケットスイッチを対象とする。それら 2 つのアーキテクチャに対してパケットスケジューリングアルゴリズムを適用してシミュレーションを行い、コストの観点からアーキテクチャの性能比較を行った。その結果、それぞれのアーキテクチャの特性、および有効性を示すパラメータ領域が存在することを明らかにし、共有バッファ型スイッチについては、負荷が高い場合パケット間の空き領域がスイッチの性能を大きく低下させることが分かった。また、本稿ではパケット間空き領域低減手法を提案し、この手法によって高負荷時においても出力バッファ型スイッチを上回る性能を示すことを明らかにした。

キーワード WDM, フォトニックパケットスイッチ, FDL バッファ, 可変長パケット, パケットスケジューリングアルゴリズム

Scheduling Algorithm with Void Space Reduction
in Photonic Packet Switch

Takashi Yamaguchi†　Ken-ichi Baba††　Masayuki Murata‡　Ken-ichi Kitayama‡‡

†Graduate School of Engineering Science,
Osaka University
1–3 Machikaneyama, Toyonaka, Osaka 560–8531, Japan
Phone: +81–6–6850–6616, Fax: +81–6–6850–6589
E-mail: t-yamagt@ics.es.osaka-u.ac.jp

††Cybermedia Center, Osaka University
5–1 Mihogaoka, Ibaraki, Osaka 567–0047, Japan
Phone: +81–6–6879–8793, Fax: +81–6–6879–8794
E-mail: baba@cmc.osaka-u.ac.jp

‡Cybermedia Center, Osaka University
1–3 Machikaneyama, Toyonaka, Osaka 560–0043, Japan
Phone: +81–6–6850–6615, Fax: +81–6–6850–6589
E-mail: murata@cmc.osaka-u.ac.jp

‡‡Graduate School of Engineering, Osaka University
2–1 Yamada-Oka, Suita, Osaka 565–0871, Japan
Phone: +81–6–6879–7728, Fax: +81–6–6879–7688
E-mail: kitayama@comm.eng.osaka-u.ac.jp

Abstract　In this paper, we comparatively evaluate two photonic packet switch architectures with WDM-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. Another 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 the packet scheduling algorithm. We compare the performance 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 a 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 to the shared buffer type switch to outperform the output buffer type switch even under high traffic load conditions.

Keywords WDM, Photonic Packet Switch, FDL Buffer, Variable Length Size Packet, Packet Scheduling Algorithm

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 electronics devices.

In this paper, we study packet scheduling algorithms for the photonic packet switch. 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 [1, 2, 3, 4, 5]. 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. Although wavelength conversion requires a higher hardware cost, it results in a better performance [6, 7]. 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 paper.

In this paper, 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.

The rest of the paper is organized as follows. In Section 2, we briefly present shared buffer type 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 architectures. Conclusions and future work are summarized in Section 5.

2 Photonic Packet Switch Architectures

The photonic packet switches that we consider in this paper 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 [8].

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 [9], 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.

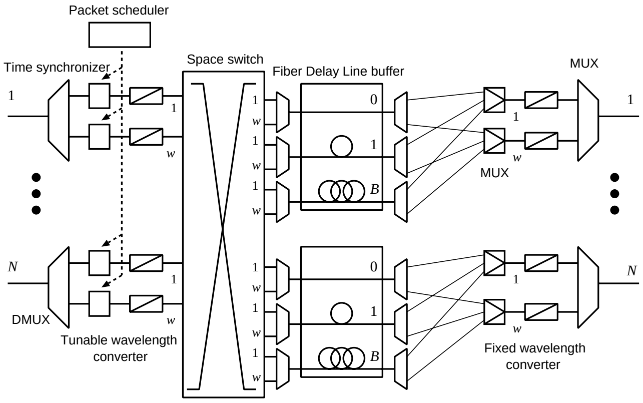

Figure 1: Output buffer type photonic packet switch architecture

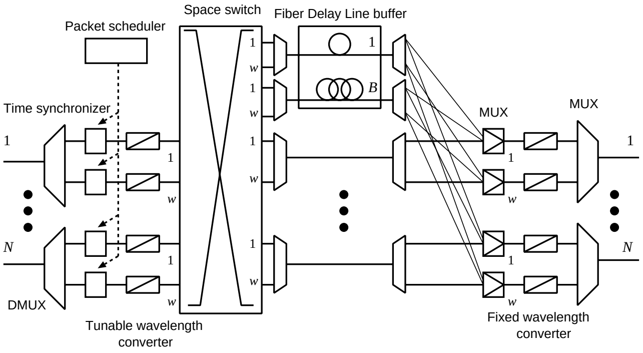

Figure 2: Shared buffer type photonic packet switch architecture

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 length of n-th delay line is n in time slot size. As we will describe later, the number of wavelengths on FDL (denoted by W_i) is equal to or larger than the number of wavelengths on the input and output fibers, W. In the following, we call the number of delay lines in one FDL buffer a *buffer depth* (denoted by B), and the number of delay lines in the whole switch a *buffer size* (denoted by B_T). The *virtual buffer size* is denoted by $B_T \times W_i$. Note that buffer depth and buffer size is identical in the shared buffer type switch, while in the output buffer type switch, the buffer size is given by the buffer depth multiplied by the number of input/output lines, as we will show below.

Figure 2 shows the architecture of the output buffer type switch, which has one dedicated FDL buffer for each output port. When the wavelengths are unused, and the packet contention can be resolved by wavelength conversion, packets are directly sent to the output ports. If several packets remain unresolved, or if there are not available wavelengths, packets are sent to FDL buffers. The $N \times N$ output buffer type switch has a number N of separate FDL buffers. The buffer size B_T is $B \times N$.

Figure 3 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. 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 [10, 11]

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 [10, 11]

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, the switch processes packets in order of arrival. Thus, when the packet is sent to FDL, a newly arriving packet with same input/output ports as the previously arriving packet should not be sent to the shorter FDL. The previous algorithms, except for Algorithm A0, have this feature. However, this feature causes the unacceptable performance degradations as we will demonstrate in the next section.

Since the output buffer type switch is equipped with the FDL buffer for every 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, using the packet scheduling method, in which the same wavelength is used for the FDL and output port, in this case 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 becomes long a high traffic load condition. Consequently the output interval between two packets destined for the same output port becomes large, and this is called the *void space* [12]. As an example, Fig. 3 illustrates why and how the void space appears, as follows. At output port 1, a packet is being 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 the port is actually in use or not.

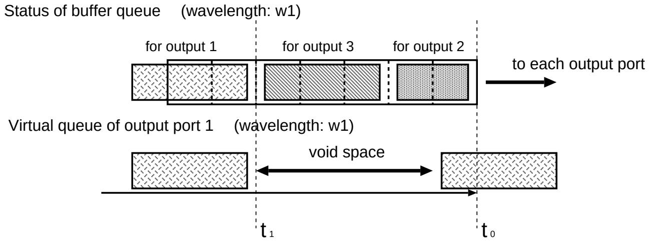
Status of buffer queue (wavelength: w1)
for output 1
for output 3
for output 2
to each output port
Virtual queue of output port 1 (wavelength: w1)
void space
t₁
t₀

Figure 3: Void space in shared buffer type switch

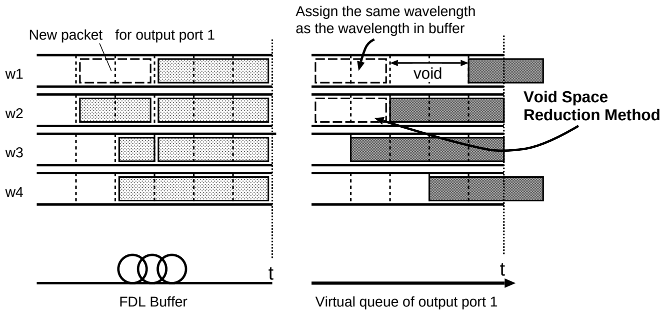
New packet for output port 1
Assign the same wavelength as the wavelength in buffer
w1
w2
w3
w4
void
Void Space Reduction Method
t
t
FDL Buffer
Virtual queue of output port 1

Figure 4: Void space reduction method

In order to solve a this type problem, a *void filling* algorithm has been proposed [12]. However, when using this algorithm, the packet scheduler needs to maintain the arriving/departing times of all packets stored in the buffer in order to insert a new packet within the void space. Therefore, the algorithm complexity is very high and is difficult to implement.

Our proposal, called the *void space reduction method*, reduces the ill–effect of the void space by using wavelength conversion. The wavelength of the packet is converted so that the influence of the void space is minimized. Figure 4 illustrates our approach. Suppose that a new packet destined for output port 1 arrives at the switch. The packet is assigned wavelength w_1 and is stored in the buffer. If the next arriving packet is assigned wavelength w_1, a 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 the void space. In the above case, therefore the new packet is assigned wavelength w_2, and thus we can avoid 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 of the combinations 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 to maintain the output time of the last packet in the virtual queue. When a new packet arrives and is decided to be 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 on the output fiber.

3.3 Computational Complexity of Packet Scheduling Algorithm

In this subsection, we discuss about the computational complexity of each packet scheduling algorithm. In the packet scheduling algorithms, the search for the optimal wavelengths in buffer or output ports and the sorting of input packets are the operations which require long calculation time. In this paper, we consider the calculation time of the search and the sorting, but ignore the processing time of other operations. Therefore, we can set the calculation time of Algorithm A0, that assigns in a round-robin fashion the wavelengths to the packets in a round-robin fashion, to 0.

First, we consider the buffer search function for assigning the optimal wavelength to the packet inserted in the buffer. In this case, since the number of the input ports is N and the number of the multiplexed wavelengths is W, the scheduler needs to search NW times in 1 slot time. Therefore, the calculation time of the buffer search function is kNW where k is the calculation time for one search in the buffer. Next, we consider the packet sorting function for the determination of the turn of the wavelength assignment to the input packets. We define the time to sort the packets into a decreasing order of packet length as m. Incidentally, the computational complexity of the sorting function is $O(NW \log(NW))$ since the maximum number of packets which arrives simultaneously is NW. Finally, we consider the output port search function for for the determination of the optimal wavelength for assigning to the packets sent to the output ports. As well as the buffer search function, since the scheduler needs to search NW times in 1 slot time, the calculation time of the output port search function is lNW where l is the calculation time for one search in the output ports.

The buffer search function is applied in Algorithm A1, A2, A3. The packet sorting function is applied in Algorithm A2 and A3. The output port search function is applied in the void space reduction method. Therefore, the calculation time of each packet scheduling algorithm is as follows. The calculation time of Algorithm A1 is kNW. The calculation time of Algorithm A2 or A3 is $kNW + m$. The calculation time of Algorithm A0 with the void space reduction method is lNW. The calculation time of Algorithm A1 with the void space reduction method is $kNW + lNW$. And, the calculation time of Algorithm A2 or A3 with the void space reduction method is $kNW + lNW + m$.

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 [9]. 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 5 and 6 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. As shown in Fig. 5, 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 Fig. 6. In the low traffic load condition, algorithm A2 again gives the best performance.

In Fig. 6, the performance of the shared buffer type switch decreases when the switch is equipped with a larger buffer

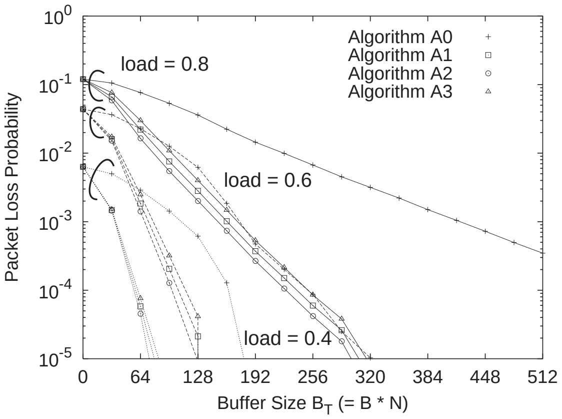

Figure 5: Packet loss probability (output buffer type switch)

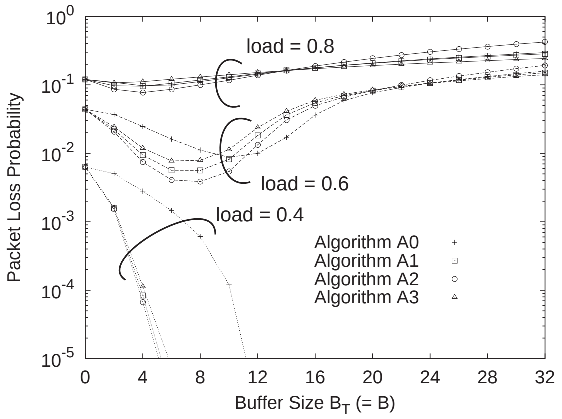

Figure 6: Packet loss probability (shared buffer type switch)

size. This is because the queue length becomes long and the possibility of a void space appearing becomes high, as was described in Section 3.3. Figures 7 and 8 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. 7, 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. 8. 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.

We evaluate the packet scheduling algorithms from a viewpoint of data loss probability, although the figures cannot be shown here for lack of space. Here, data loss probability is defined as the ratio of the total amount of dropped packets to the total amount of input packets. The simulation results show the same tendency as the previous set of figures for packet loss probability (Figs. 5 and 6), 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 9 shows the performance of the shared buffer type switch when the void space reduction method is applied. 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.

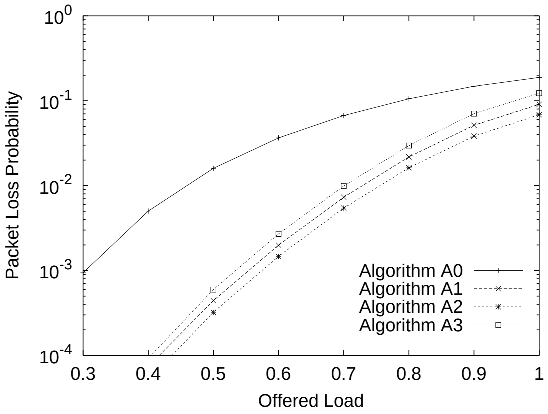

Figure 7: Packet loss probability (output buffer type switch, for $B = 64$)

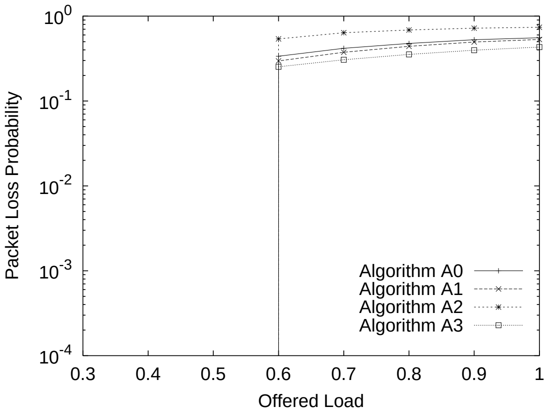

Figure 8: Packet loss probability (shared buffer type switch, for $B = 64$)

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. 10, 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. 11.

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.

5 Conclusions

In this paper, we have evaluated the performance of the shared buffer type switch and the output buffer type switch by applying packet scheduling algorithms. We have compared these two switching architectures taking into account the total number of FDLs. Our simulation results showed that the shared buffer type switch achieves a better performance than the output 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 more than that of the output buffer type switch.

In future work, we need to evaluate the hardware cost more precisely. And, we need to evaluate the performance of switches using more and better metrics.

Acknowledgments

This work was supported in part by TAO (Telecommunications Advancement Organization of Japan) under the project “Research and Development Project of Photonic Router for High Speed, High Quality, and Highly Efficient Internet ”.

References

[1] 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, February 1997.

[2] 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.

[3] D. Hunter, M. C. Chia, and I. Andonovic, “Buffering in optical packet switches,” *IEEE Journal of Lightwave Technology*, vol. 16, pp. 2081–2094, December 1998.

[4] 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, March 1998.

[5] S. Yao, J. Zhang, B. Mukherjee, S. Dixit, and S. J. B. Yoo, “PLATO: A generic modeling technique of all-optical packet-switched networks,” *to be presented at INFOCOM 2002*, June 2002.

[6] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, “All-optical packet-swtiched networks: A study of contention-resolution schemes in an irregular mesh network with variable-sized packets,” in *Proceedings of OptiComm 2000*, pp. 235–246, November 2000.

[7] 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, February 1997.

[8] M. Murata and K. Kitayama, “Ultrafast photonic label switch for asynchronous packets of variable length,” *to be presented at IEEE INFOCOM 2002*, June 2002.

[9] F. Callegati, “Optical buffers for variable length packets,” *IEEE Communications Letters*, vol. 4, pp. 292–294, September 2000.

[10] 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, November 2000.

[11] F. Callegati, W. Cerroni, and G. Corazza, “Optimization of wavelength allocation in WDM optical buffers,” *Optical Networks Magazine*, vol. 39, pp. 66–72, November 2001.

[12] 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, October 2000.

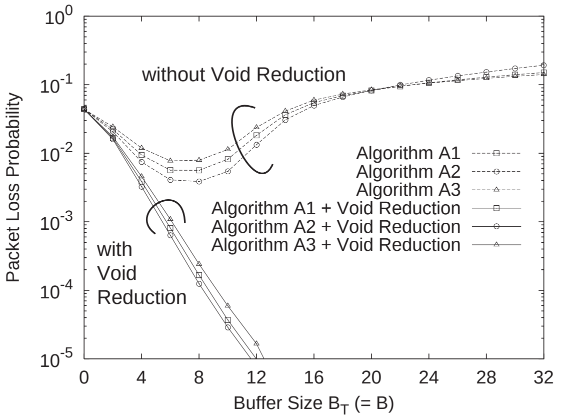

Figure 9: Packet loss probability (shared buffer type switch, void space reduction method, load = 0.6)

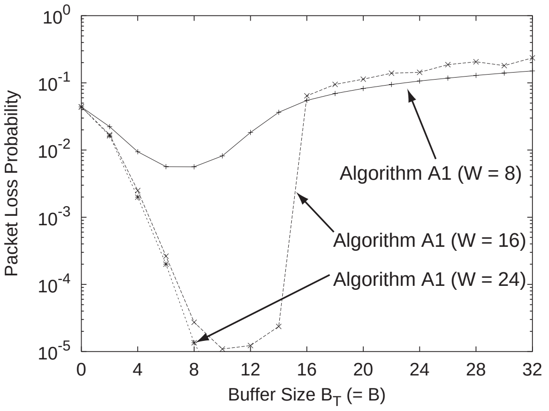

Figure 10: Packet loss probability (shared buffer type switch, increasing the inner wavelengths, load = 0.6)

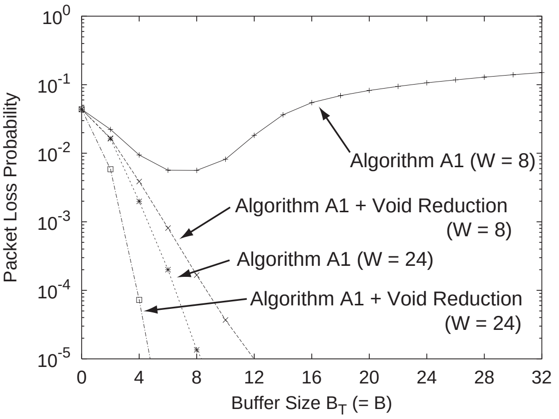

Figure 11: Packet loss probability (shared buffer type switch, void space reduction method + increasing the inner wavelengths, load = 0.6)