Ultrafast Photonic Label Switch for Asynchronous Packets of Variable Length

Masayuki Murata* and Ken-ichi Kitayama**

* Cybermedia Center, Osaka University
Toyonaka, Osaka 560-0043, Japan
murata@ics.es.osaka-u.ac.jp
** Graduate School of Engineering, Osaka University
Suita, Osaka 567-0871, Japan
kitayama@comm.eng.osaka-u.ac.jp

***Abstract* - This paper describes new optical switching architectures supporting asynchronous variable-length packets. Output line contention is resolved by optical delay line buffers. By introducing a WDM technology, parallel buffer can be equipped with multiple wavelengths on the optical delay line buffer. Using an ultrafast photonic label processing technique, an implementation of our architecture would be fast enough for packet scheduling that selects the appropriate output port, wavelength, and delay line buffer. To evaluate the switch performance, we model an output port of our switch as a multi-server and multi-queue system where each server corresponds to a wavelength and where each arriving packet joins the shortest queue. We use an approximate analytic approach to evaluate the switch performance. The results of the analysis and of simulation experiments show that the use of the WDM technique can greatly improve the switch performance in terms of packet loss probabilities.**

***Index terms* - Photonic Switch, Photonic Label, WDM (Wavelength Division Multiplexing)**

I. INTRODUCTION

There are two critical issues we need to address if we are to make an ultrafast optical switch that can handle asynchronous packets whose lengths can vary. The two major bottlenecks with regard to switch performance are the looking up of the next hop in the forwarding table and the packet buffering for contention resolution. The first bottleneck in electronic routers is the longest-prefix-match for each incoming packet. The speed of the lookup algorithm a routing table uses is determined by the number of memory accesses required in order to find the matching entry and by the memory speed. The memory access time typically ranges from 10 to 60 ns. If an algorithm performs eight memory-lookups with a memory access time of 10 ns, only 12.5 million look-ups can be performed in one second [Kes98]. For a 800-bit long packet, for example, the bit rate of the link interface with the router is only 10 Gbps. This processing capacity is much smaller than the aggregate capacity of the wavelength-division-multiplexed links in even one optical fiber. For example, 40 Gbps x 160 wavelengths = 6.4 Tbps per fiber [Ito00]. This means that the rough estimation that, for comparably priced hardware, switching speeds are 20 times greater than forwarding speeds [Lin97] holds even for photonic technologies.

One promising way to increase the capacity of routers is to use MPLS (Multi-Protocol Label Switching) technology [Dav98]. This technology separates switching and forwarding functions so that full use can be made of the high-speed switching capability of the underlying network. The packet-forwarding functions needed to determine the destination are performed only at the edge of the MPLS domain. While MPLS needs to establish a closed domain for utilizing a new lower-layer technology, it is useful to incorporate the photonic technology for building the very high-speed Internet. There are, however, still several problems the need to be solved if we are to deploy MPλS. The most difficult problem is capacity granularity: the unit of the bandwidth between the edge node pairs of the MPLS domain is a wavelength capacity. It may sometimes be too large to accommodate the traffic between node pairs. One approach to resolving the capacity granularity problem is addressed in [Ban00], where the authors introduce wavelength merging, but the related technology is still immature.

Another promising technique is to utilize a recently developed photonic label switching technology described in [Kit99]. It enables bandwidth efficiency to be increased by resolving the granularity problem in an optical domain. A photonic label is attached to the head of the payload data (Figure 1), and a family of optical code sequences originally used as signature codes in optical code division multiplexing (OCDM) [Pru86], [Sal89] is used as the photonic labels. The recognition of a specific code sequence is accomplished by correlation in the optical domain. As this can be done simply by using a passive optical waveguide device, the recognition time is governed by the propagation delay of the device. This is a key to

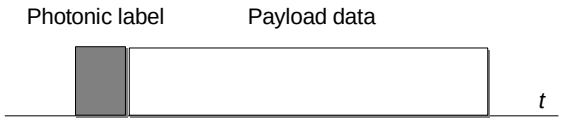

Figure 1: Structure of an optical packet with a photonic label.

ultrafast photonic label processing [Kit99], which is superior to the existing approach utilizing the optical code as in [Mea00] where the authors employ a sub-carrier-multiplexed (SCM) optical header. There is another optical packet switching method called optical burst switching [Qia99], but it requires estimation of delays at each node and for each path. A MPLS technology utilizing the photonic label switching method is abbreviated *OC-MPLS* in this paper.

The second bottleneck is caused by queuing and buffering to resolve the contention at the output ports of the switch. We need to develop an optical packet switch resolving the contention at the output ports. We can do this by adding a fiber delay line for buffering the packets failing to acquire the output line. Performance should improve if WDM is used so that the optical delay line buffer can be shared, but for that we need a scheduling algorithm putting each packet into the appropriate queue. Fortunately, photonic label switching can be used not only to process packets at a very high speed within the packet switch but can also be used to handle IP packets directly. That is, our switch can handle asynchronously arriving packets with variable lengths.

Research efforts have been devoted to optical packet switches recently, but most have been devoted to the switches handling fixed-size packets. See, e.g., [Hun98]. Few take the approach described in [Tan00], where the authors describe an optical packet switch handling variable-length asynchronous packets. To resolve output contention at the delay line buffer, we developed a scheduling algorithm based on a concept of a void filling. The authors of [Ge00] consider the scheduling policy for storing simultaneously arriving packets in an optical buffer with different wavelengths, and they compare four scheduling policies in terms of packet loss probability. A major problem with the existing approaches is that the packet header processing is assumed to be performed in an electronic domain, where complicated scheduling is likely to be a bottleneck. We therefore carefully consider the implementation issues relevant to the very high-speed packet switching in our switch.

The remainder of this paper is organized as follows. Section II describe our switch architecture handling asynchronous and variable-length packets in an optical domain, Section III discusses implementation issues in detail, Section IV describes the results of approximate analysis and simulation experiments evaluating the performance of our switch, and Section V concludes our paper by briefly summarizing it and mentioning future research topics.

II. SWITCH ARCHITECTURE

Here we describe two new switch architectures, one with wavelength conversion and the other without it. The hardware is simpler when wavelength conversion is not allowed, but we get better performance when it is.

A. Optical Switch without Wavelength Conversion

The optical switch without wavelength conversion consists of three optical units: a switching unit, a scheduling unit, and a buffering unit. A 2x2 optical switch is illustrated in Figure 2, where the number of wavelengths is designated by *W*. As shown in the figure, each component is dedicated to a single wavelength channel.

The switching unit switches packets according to the information in the photonic label. If the packet is destined for the output port O_1, the switch is set to the bar-state, directing the packet to the upper part of the optical switching unit. This switching can be done by processing the photonic label, and the label recognition time is determined by the propagation delay of the optical decoder device, making the photonic label processing ultrafast [Kit99]. According to the experimental results reported by [Wad00], it should be possible to process the photonic labels of more than 10^9 packets in one second.

The buffering unit provides an optical buffer by using fiber delay lines. Let *D* be the delay of a fiber delay line (which in this paper we will call the *unit delay*). Then a packet to be delayed for an interval *iD* is put on the *i*-th delay line (shown in the figure by τ_i). The counter b_{ij} keeps the buffer status information for wavelength λ_j going to the output port O_i. This switch can handle variable-length packets because whenever a packet arrives at the optical buffer, this counter is incremented as follows:

$$b_{ij} \leftarrow b_{ij} + \lceil x/D \rceil \qquad (1)$$

where *x* denotes the length of the arriving packet. And every *D* it is decremented by one and the next arriving packet is put on the b_{ij}-th delay line.

The heart of our optical switch is the optical scheduling unit. In the switch without wavelength conversion, each optical scheduler S_{ij} in this unit is dedicated to

scheduling packets on wavelength λ_j that are destined for the output port O_i. It reads the packet length in the header of the arriving packet, and updates the buffer status b_{ij} according to Eq. (1). Then after the time synchronization, each bit of the payload in the optical packet is encoded with an optical code—the structure of which is the same that of the photonic label shown in Figure 1 and used outside the photonic label switch to indicates the specific delay line into which the packet is to be fed—and the encoded packet is split and delivered to the optical decoders, only one of which recovers the data bits of the packet. Then the packet is transferred to an appropriate delay line. See Subsection III-A for the optical buffer assignment based on photonic label processing.

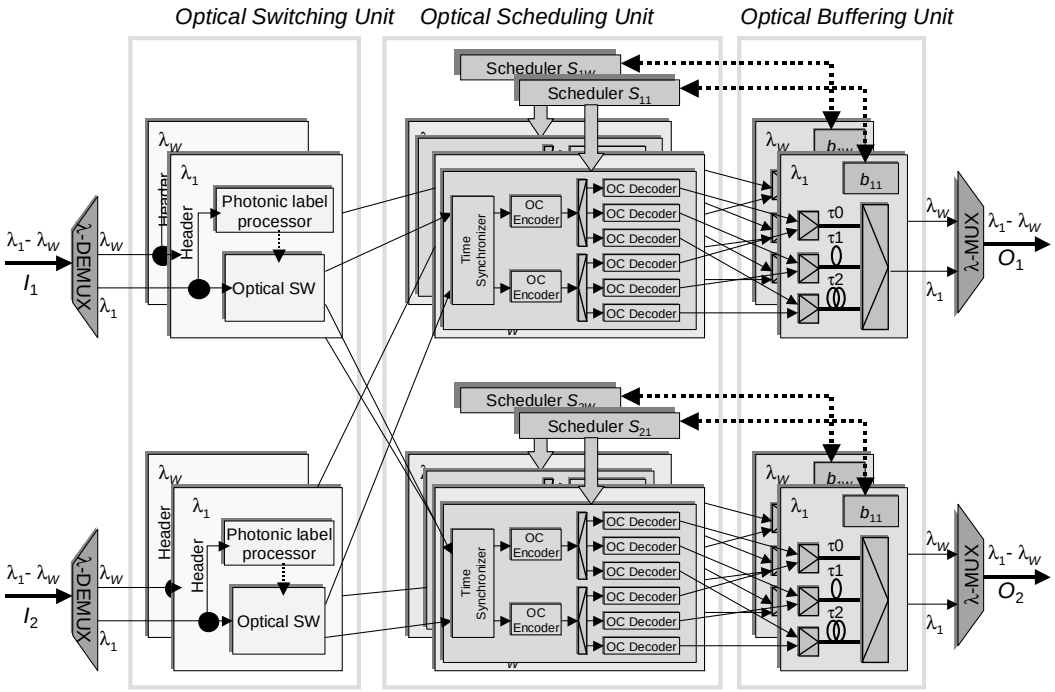

Figure 2: Optical switch without wavelength conversion.

One problem is that we need to handle packets arriving simultaneously from different input ports. Each scheduler ensures that the counter stays valid by perform the following three-step operation for each packet: (1) read the buffer status information, (2) update it according to the packet length, and (3) write it back into the memory. Because it must not receive another packet during this three-step operation, a time synchronizer delay an arriving packet if the scheduler is processing another packet. Although there needs to be only one time synchronizer for each input port, since from any given input port only a single packet arrives at one time, the introduction of time synchronization increases the hardware complexity. All incoming packets destined for the output port O_i should be processed in sequence. See Subsection III-C for more detail.

B. Optical Switch with Wavelength Conversion

The switch with wavelength conversion has wavelength converters, indicated in Figure 3 by supercontinuum light source with gate switches (SC + Gate), added to the optical scheduling unit. The wavelength conversion allows the WDM buffer to be incorporated into the switch fabric. In this case, a packet in contention will be put on an alternative delay line buffer by changing the wavelength in front of the buffer. See Subsection III-B for a novel wavelength conversion method. This wavelength conversion is expected to reduce packet loss significantly, and this reduction will be discussed in more detail in Section IV.

The optical scheduling unit of the switch allowing wavelength conversion is a little more complicated that that of the switch not allowing. To schedule the packets destined for the output port O_i, the scheduler S_i of the output port O_i has to know the status of three delay lines τ_0, τ_1, and τ_2 with W different wavelengths. When a packet arrives, the scheduler searches for the shortest queue, which can be easily determined by checking the counters b_{i1} through b_{iW}. This can be done by a simple hardware comparator. Once the scheduler determines the wavelength to be tuned (say, λ_j), it updates the counter b_{ij} and sets the gate for the selected wavelength.

III. Optical Implementation Issues

A. Optical Buffer Assignment Based on Photonic Label Processing

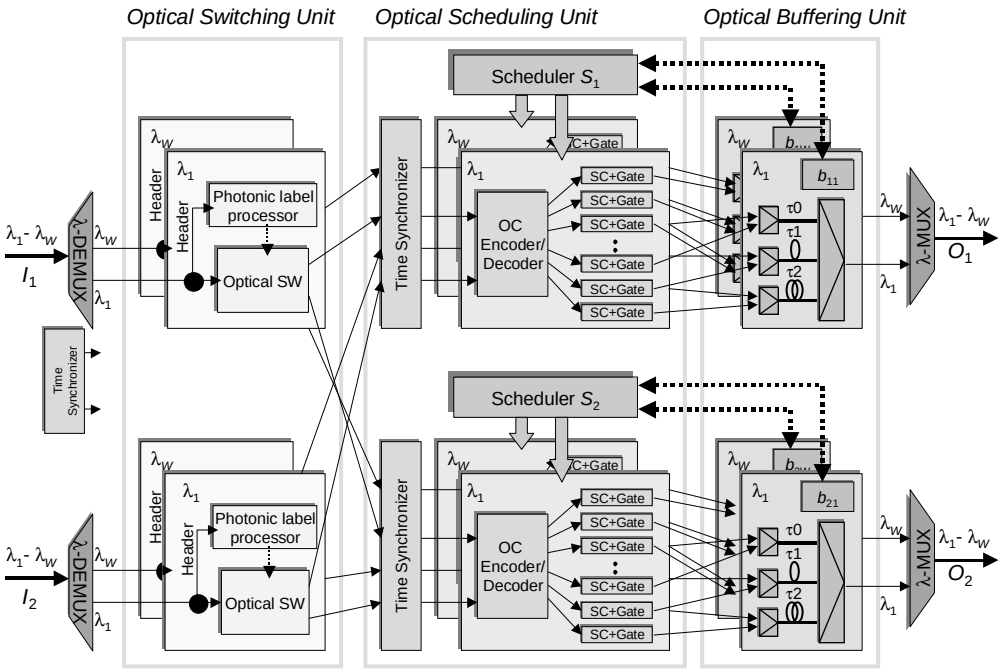

Figure 3: Optical switch with wavelength conversion.

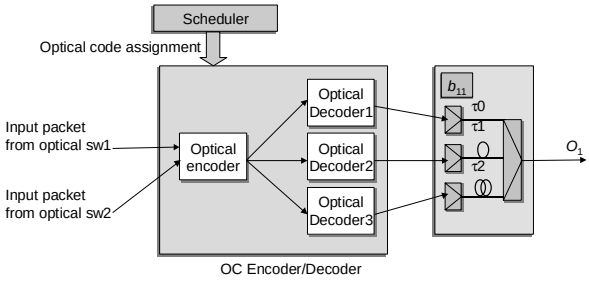

Figure 4: Buffer assignment in the optical domain.

Photonic label processing is also exploited within the switch for buffer assignment. The optical codes used in the buffer assignment are the same as those used as the photonic labels for network-wide OC-MPLS outside the photonic switch, but they are of significance only within the switch. Therefore the scheduler has to be provided only with the information about the buffer status within the switch. Figure 4 illustrates the schematic of proposed buffer assignment in the optical domain. For simplicity, suppose that the buffer has three delay lines, τ_0, τ_1 and τ_2, with different lengths as shown in Figure 4. The packet from the output port of the switch is encoded with a photonic label which designates an appropriate delay line to be fed and is delivered to optical decoders 1 through 3. The scheduler determines the appropriate delay line for the packet to be fed to and assigns a photonic label. Because the packet is marked with the photonic label, the output emerges only from the decoder assigned the same label as the incoming packet. Note that each bit of the payload is encoded with the photonic label, while in OC-MPLS the photonic label is used only in the header (Figure 1) and the payload data bits are not encoded. The output from the decoder recovers the original bit sequence of the payload, and the recovered packet is fed into a desired delay line, resulting in contention resolution in the optical domain.

Figure 5 shows a special class of optical encoder/decoders [Wad99]. The incoming pulse stream (from the upper *l.h.s*) is encoded into 8-chip bipolar phase-shift keying (BPSK) optical code through the tapped delay line, followed by the optical phase shifter (from the *r.h.s* on the bottom). The optical carrier of the split pulse is phase-shifted by 0 or π, resulting in the bipolar phase-shift keying. Both the tapped delay line and the phase shifter are tunable, retaining the programmability of optical codes. The decoding is carried out with the same device. The optical code launched from the upper *r.h.s.* is correlated, and the autocorrelation peak emerges if the assigned code matches that of the incoming optical code (from the *l.h.s* on the bottom). Obviously, the time that optical correlation takes is equal to the propagation time of the code in the decoder. In the decoder version of the device illustrated in Figure 5, the chip pulse at the tail passes through only 70 ps after the leading chip pulse enters the device. This speed is the key to ultrafast photonic label processing. It is noteworthy that all the process is carried out in the optical domain, and the photonic label is recognized without any logic operation. If we make the number of delay lines (i.e., the buffer size) 10, five chips are enough for 2^4 optical codes. A buffer size of 10 is actually a reasonable design choice, as we will discuss in Subsection IV-D.

Since photonic label recognition using 8-chip BPSK optical codes has been demonstrated at 10 Gbps [Wad00], it should be possible to increase the bit rate for photonic label recognition using 5-chip BPSK optical code to 100 Gb/s by using the readily available

1-ps optical pulse. Compared with the optical code in Figure 5, the chip pulse interval is shrinked from 5ps to 1ps.

B. Supercontinuum Wavelength Converter

This subsection focuses on the switch architecture in Figure 3 and describes devices for the wavelength conversion between the optical decoder and the optical delay line buffer. WDM buffering can be used to share the delay line with packets of different wavelengths. For example, when the packet from the decoder on the λ_1 plane fails to find an empty delay line in b_{11} but finds a non-empty delay line τ_0 on b_{1k}, it can be fed to the non-empty delay line after its wavelength is converted to λ_k. Note that the group delay time difference caused by the wavelength difference is negligible.

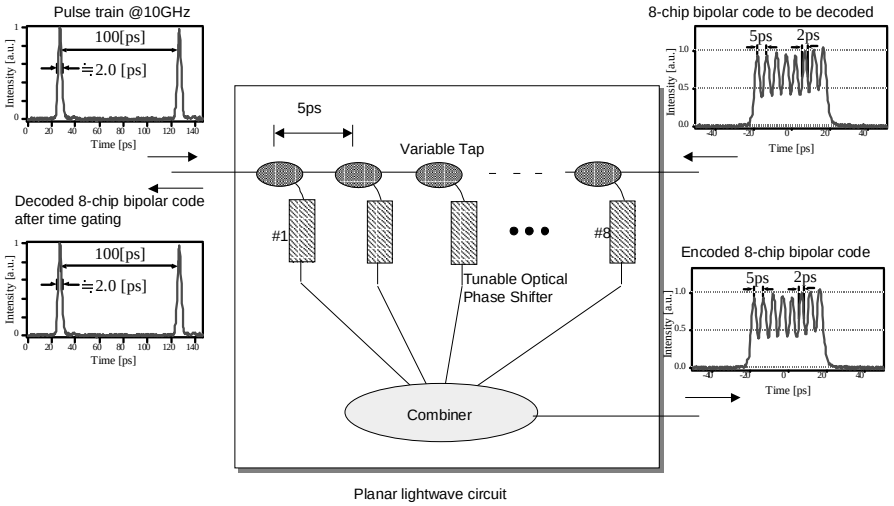

Figure 5: Optical encoder/decoder.

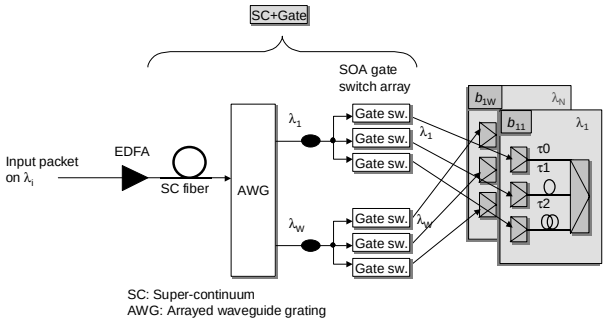

Figure 6: Structure of wavelength conversion unit.

Here we describe a new wavelength conversion scheme that does not use the readily available semiconductor optical amplifier (SOA) tunable wavelength converter [Dan98]. Although we must not neglect the practical availability of SOA wavelength converters, we focus here on a novel device more suited to tomorrow's ultrahigh-bit-rate dense WDM (DWDM) systems. As shown in Figure 6, it consists of a supercontinuum (SC) fiber combined with a wavelength demultiplexer (AWG) and a SOA gate switch array (gate switch). Compared with conventional SOA wavelength converter, its advantages include high-speed gate switching, a large number AWG output ports, and the ultrafast SC response due to the nature of the fiber nonlinearity and the ultrawide spectrum of the emission. Disadvantages, however, are its larger number of components (by a factor of 3*W*, where *W* is the number of multiplexed channels) and the additional optical amplifiers needed so that the input signal can serve as the pump.

A supercontinuum pulse source generates picosecond pulses at several tens of gigabits per second over an extremely broad spectral range. The spectrum of the seeding short pulse at a specific repetition rate is broadened because of the fiber nonlinearities, so using the AWG to pick desired wavelength components out of the SC spectrum makes it possible for the single light source to serve as the source of pulses with various wavelengths. Another advantage is that it uses only one pump laser, and its fixed channel spacing with accuracy equivalent to that of a microwave oscillator enables the entire chain to be locked to an absolute standard by locking just one mode of the chain. A typical SC spectrum, both the temporal waveforms and the spectra of the spectrum-sliced output, is shown in Figure 7 [Sob98]. Recently, 1,010-channel with the frequency interval of 12.5GHz at 2.5Gbit/s over 100nm in 1550nm spectral region has been achieved [Tak00].

C. Time Synchronization

Our switch uses a time synchronizer to ensure that a packet does not enter the optical switching unit before the scheduler finishes directing the preceding packet to the buffer. Located in front of the optical OC encoder in the optical scheduling unit, the time-synchronizer aligns the incoming packet to the time-slot as shown in Figure 8. Incoming packets therefore wait at the input port until the scheduler of that port completes the task for the preceding packet. Note that the synchronization is local (within the switch) and is independent of the global clock that the

electrical interface of the switch has, thus maintaining asynchronous operation at the bit level. Also note that the slot duration is set equal to the processing time of the scheduler for a packet and is much longer than one-bit duration of the clock. If, for example, the processing time of the electronic scheduler is roughly 2 ns and the time resolution of the synchronization has to be less than 10% of the time slot, the maximum delay time required for the time synchronizer becomes 2 ns with the time resolution of less than 0.2 ns.

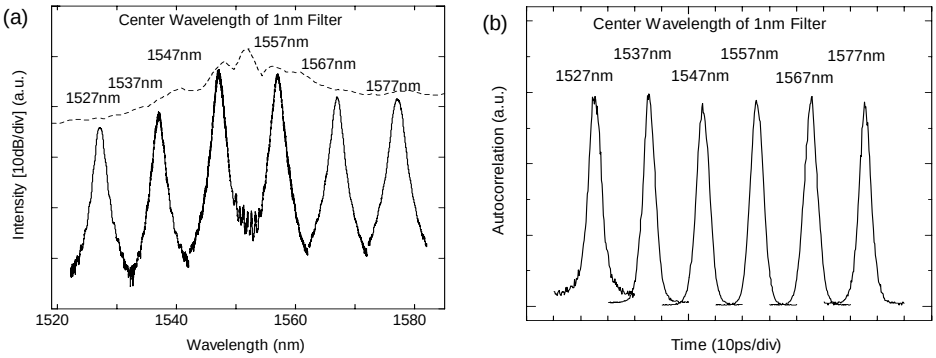

Figure 7: Typical supercontinuum (SC) spectrum at a repetition rate of 10 GHz; (a) optical spectrum of spectrum-sliced WDM channels, (b) typical waveforms of peaks in the spectrum of spectrum-sliced WDM channels.

The block diagram of the time synchronization is depicted in Figure 9. The process includes time alignment, start recognition, and time evaluation. The start time of the packet is determined from the packet header, and the necessary delay time is calculated. Only the time alignment has to be carried out in the optical domain; the start recognition and time evaluation can be done quickly enough by electronics. The time alignment can be accomplished by using a variable delay line, and there have been several synchronization schemes for "coarse" and "fine" time alignments, respectively using a switched delay line [Chl96] and group delay dispersion combined with the wavelength conversion [Fra99]. A suitable device would be the cascaded switched delay line shown in Figure 10. The lengths of the delay lines are arranged so that the first delay line is equal to 1/2 the time slot duration, the second one is equal to $1/2^2$ the time slot duration, and so on. Each 2x2 optical switch between the delay lines is configured with the correct path based upon the time evaluation. Four-cascaded delay lines, the first of which is 200 mm long (corresponding to 1 ns) can guarantee a time resolution of less 0.2 ns.

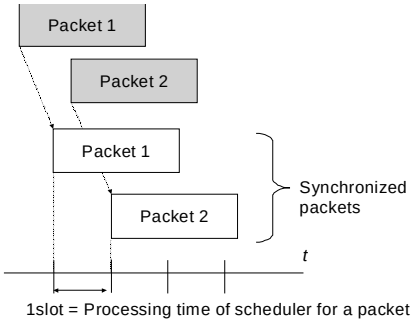

Figure 8: Time synchronization of packets.

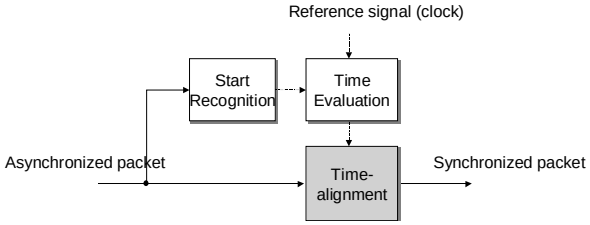

Figure 9: Block diagram of time synchronization.

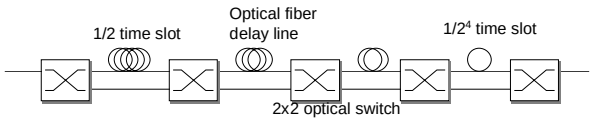

Figure 10: Cascaded delay line buffer.

IV. Performance of Proposed Optical Switch

Our analysis was rather simple and, for example, did not consider the time synchronizer described in the previous section. This simplicity, however, implies that our analysis approach and results are broadly applicable and thus that our discussion here can be applied to the other optical switches utilizing the fiber delay lines as a packet buffer.

A. Analysis Approach

To evaluate our optical switch, we focused on one output port since the switch is a non-blocking one. In the switch without wavelength conversion, packet switching is performed on each wavelength independently. The packets arriving at the input ports are switched in the optical switching section to the designated output port. Thus it can be modeled as a single queue, where the server corresponds to one wavelength. In this case, our concern is the influence of introducing the fiber delay line as the packet buffer, which is stud-

ied in [Cal00]. When the packet arrives and the server is idle, the packet is transmitted immediately. If the server is busy, on the other hand, the packet is queued. Let t be the arrival time and t_f be the time at which the server will be free to serve the new packet. In the case of electronic buffers, the new packet can be served after $t_f - t$. In the case of fiber delay line buffers, however, we need to consider the "granularity" of the delay lines. That is, when the fiber delay line buffers the packet, the buffering time is measured in units of the fiber line delay D, and therefore only a finite set of delays can be achieved. The new packet is delayed by an amount given by the following equation:

$$\Delta = \left\lceil \frac{t_f - t}{D} \right\rceil D \qquad (2)$$

In other words, an excess length of $\Delta - t_f + t$ is additionally brought to the server in the case of fiber delay line buffers.

Based on the above observation, Callegati introduces an additional service time for each packet if the packet finds the server to be busy [Cal00]. In the queuing system, which can be described by a birth-and-death process, the author presents an iteration algorithm to find the packet loss probability. The granularity D affects the performance as follows: if D is small, the time resolution of the fiber delay line is large and decreasing D therefore improves performance but reduces the buffer capacity (in bytes). Increasing D, on the other hand, increases the buffering capacity but reduces the time resolution of the buffer becomes small and introduces a larger excess load. Callegati finds through numerical examples that when the average packet size is one, the optimal value of D is about 0.3 irrespective of the buffer size. We note here that the number of delay lines corresponds to a buffer size we call *buffer depth* here and represent by B. The buffer capacity in bits is then determined as $B \times D$.

For the switch allowing wavelength conversion, we have multiple servers, each of which has a dedicated queue (Figure 11). Our scheduling policy is to place the packet in the shortest queue if all servers are busy, and in our switch this is done by choosing the smallest counter. This kind of system is studied in [Lin96], in which the authors studied a multi-server and multi-queue system with a "Join the Shortest Queue" (JSQ) policy. In our case, the server corresponds to the wavelength and has a buffer consisting of optical delay lines.

Our approximate analysis of our system is one with two parts, the first of which extends the approximate analysis developed in [Lin96]. Since the buffer capacity is assumed to be infinite in the original approach, we introduce another approximation to treat the finite buffer capacity. The second part of our analysis uses the approach in [Cal00] to take account of the granularity of the fiber delay lines. Since our approximate analysis is rather straightforward, we summarize our approach in Appendices A and B.

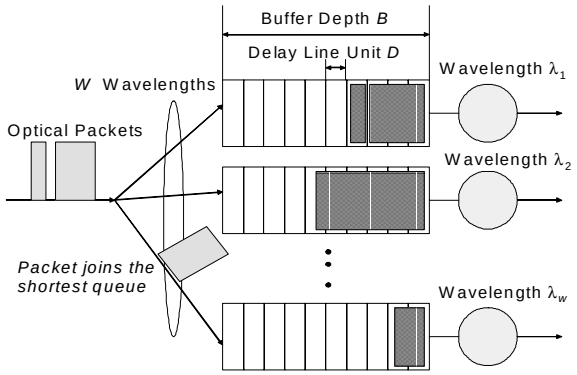

Figure 11: Model for an optical delay line buffer utilizing WDM.

B. *Effects of Multiple Wavelengths*

This subsection presents numerical results obtained by applying our analysis. The traffic load ρ per wavelength is given by

$$\rho = \lambda/(\mu W) \qquad (3)$$

where λ is the total packet arrival rate at the output port and μ is the inverse of the average packet length. In numerical examples, the packet arrival rate λ is set proportional to the number W of wavelengths and the traffic load per wavelength is fixed. For simplicity of presentation we set the average packet length to be unity and set D to be relative to the average packet length.

The effect of the unit line delay D on packet loss probability is shown for various numbers of wavelengths in Figure 12. The analytical results are shown by solid lines, and simulation results for W = 1, 2, 3, and 4 are also shown so that the accuracy of our approximate analysis can be assessed. (In simulations we generated a billion packets.) The traffic load ρ is fixed at 0.8 and the buffer depth B is 64. Note that the case of W = 1 corresponds to the result in [Cal00]. From the figure we can see that the optimal value of D is close to 0.3 irrespective of the number of wavelengths. We can also see that increasing the number of wavelengths can dramatically improve the switch performance if D is selected appropriately. In the current example setting, the traffic load per wavelength is identically set. This implies that the case of W = 1 corresponds to the switch without wavelength conver-

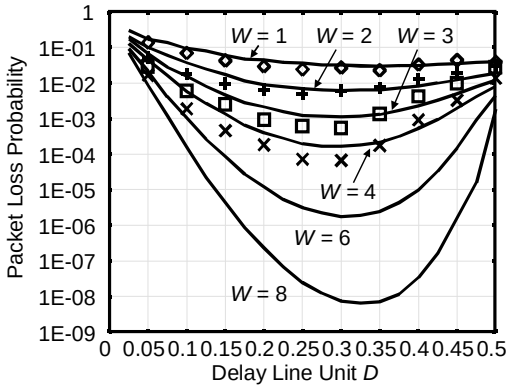

Figure 12: Packet loss probability as a function of delay unit D.

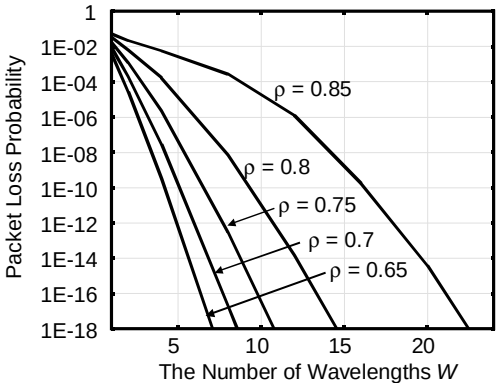

Figure 13: Packet loss probability as a function of the number W of wavelengths.

sion, in which each of multiple wavelengths forms an independent queue. Even the very high packet loss probabilities in the case of $W = 1$ are rather optimistic because our analysis implicitly assumes that the traffic load is well balanced among independent wavelength channels of W.

The effect of introducing the wavelength conversion can be seen in Figure 13, where for various traffic loads the packet loss probability (when the buffer size is 64) is shown as a function of the number of wavelengths.

Buffer size is an important design parameter because providing a number of delay lines for each output port directly affects the switch cost, and the dependence of packet loss probability on B is shown in Figures. 14 and 15 for $W = 1$ and $W = 8$. As can be seen in Figure 14, a quite large amount of the buffer (number of delay lines) is needed to decrease the packet loss probabilities in the switch without wavelength conversion. And we can see by comparing Figure 15 with Figure 14 that when the number of wavelengths is increased from 1 to 8, we need only about one-tenth as much buffer capacity to get the same packet loss probabilities for given traffic load per wavelength.

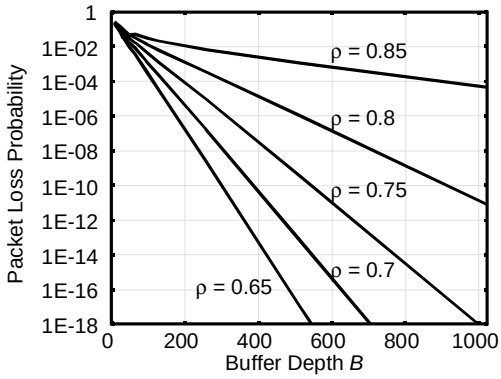

Figure 14: Packet loss probability as a function of buffer depth B when $W = 1$.

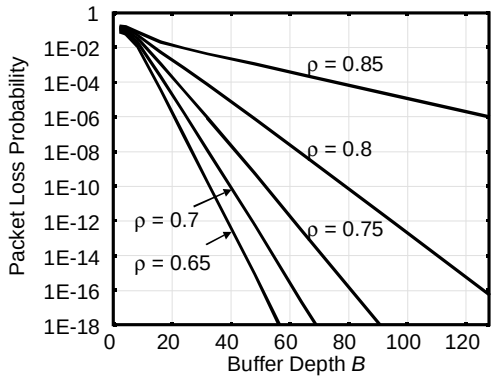

Figure 15: Packet Loss Probability as a function of buffer depth B when $W = 8$.

C. *Influence of Packet Size Distribution*

Because the distribution of Internet packet sizes is not an exponential distribution, we investigated the influence of the packet size distribution on switch performance. For this purpose, we used the actual traced data found at [WAN97]. See Figure 16, where the exponential distribution with same mean is also plotted for reference. Note that in the figure, we omit very small probabilities for the packet with about 4,000 bytes. Since it is difficult to evaluate the switch performance analytically, we used simulation experiments. For consistency, we set the average packet size to be unity (corresponding to 257.1 bytes in the current traffic data). Since we used a simulation, the parameter region for the small packet loss probabilities could not be examined. The results, shown in Figure 17, differ from those obtained when we assumed exponentially distributed packets (Figure 12). The packet loss probabilities are high for packet sizes around 500 bytes (Figure 16), leading to the large packet loss probabilities seen around $D = 0.5$ for the actual (nonexponential) distribution. Except for this difference, though, it can be seen that the effect of the packet size distribu-

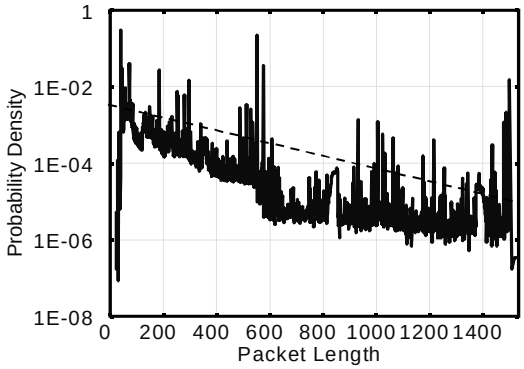

Figure 16: Packet size distribution (data from [WAN97])

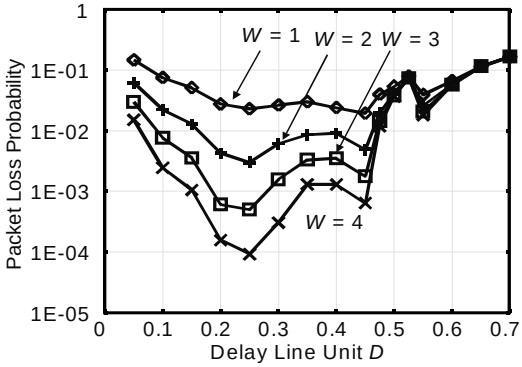

Figure 17: Packet loss probability as f function of unit delay *D* (assuming the packet size distribution shown in Fig. 16)

tion is not large, and we can expect that the analysis results presented in the previous subsection hold even for the actual packet size distribution with very small packet loss probabilities. It is clear that the best results are obtained when the unit line delay *D* is about 0.25.

D. Optimizing the Number of Wavelengths and the Buffer Size

We last discuss the optimization of the WDM buffer. There will be optimum values for the number of wavelengths and number of delay lines needed to satisfy a required packet loss probability. One problem with a large WDM buffer would be a skew, which is the group delay difference caused by the group delay dispersion of the fibers. When the packets of different wavelengths are fed into a delay line, two or more packets emerge at different timing, complicating the administration by the schedulers. For 10 and 100 wavelengths, the group delay differences for various buffer sizes are summarized in Table 1. These values were calculates assuming that the channel spacing is 1 nm and that the delay lines are standard single-mode fibers having a dispersion of 17 ps/nm/km at the center wavelength of λ = 1550 nm. Note that a unit length of the delay line is set to be equal to the average packet size of 2,000-bit at the bit rate of 100 Gb/s and D=1.0 in calculation.

Table 1: Skew [ns] for various buffer sizes and numbers of wavelengths.

Wavelengths	Buffer size *B* [in packets]		
	10	100	1,000
10	$6.8\text{x}10^{-3}$	$6.8\text{x}10^{-2}$	$6.8\text{x}10^{-1}$
100	$6.8\text{x}10^{-2}$	$6.8\text{x}10^{-1}$	6.8

We can see from the values listed in Table 1 that the skew will be prohibitively large when there are more than 100 WDM channels and the buffers are large enough to hold more than 1,000 packets. Considering the tolerable range of the skew will be within the 10% of the packet length, it becomes 0.4m. Note that 2,000-bit long packet at the bit rate of 100Gb/s is 4m. Therefore, 100 WDM channels with the buffer size of 1,000-packet may not be allowed. Another problem with a long optical fiber delay line is that the dispersion effect and the optical loss distort the waveform of each optical pulse. Dispersion compensation fibers and optical amplifiers [Hal99] therefore have to be used, and they make the buffer rather complicated.

V. CONCLUDING REMARKS

In this paper we have described a new optical switch that is based on the photonic label switching techniques and can handle asynchronous packets of variable lengths. Contention resolution at the output buffer is resolved by using fiber delay lines as packet buffers. Incorporating the WDM technology into the optical delay line buffer dramatically improved the switch performance. It has been shown by a newly developed approximate analysis method that the WDM buffer assignment can be implemented using ultrafast photonic label processing. We have also tested the case of generally distributed packets. Another important case is related to the packet arrivals; i.e., a heavy-tailed distribution might be necessary for actual buffer dimensioning [Tan00].

Our switching architecture is intended to be used in MPLS-based networks. Its advantage is that the packet loss can be well dimensioned by the traffic engineering approach [Dan00], but it requires building a closed cloud of MPLS networks. Another possibility is that our proposed switch be used as IP routers are. Since there is no restriction on the content of a photonic label, it can be an IP address. A longest prefix matching is also allowed by the current photonic

label processing technology [Kit99].

APPENDIX A: ANALYSIS OF A MULTI-SERVER MULTI-QUEUE WITH A "JOIN-THE-SHORTEST-QUEUE" POLICY

In this appendix we develop an approximate analysis for our optical switch with wavelength conversion. The operation of our switch can be modeled by a multi-server and multi-queue system where each server is equipped with finite buffer and where a newly arriving job that finds no idle servers is buffered at the shortest queue. Here the server corresponds to the wavelength, and the job corresponds to a variable-length packet.

Since the buffer is implemented by the fiber delay line, we need to consider the granularity of the delay unit because it affects the switch performance. This was done by Callegati [Cal00], who considered a single queue governed by the birth and death process. That is, a single wavelength is treated in [Cal00].

Our system, in contrast, has *W* wavelengths and the delay line can be shared by those wavelengths. It leads to the above-mentioned multi-server and multi-queue system. In [Lin96], the authors treat such a system with infinite buffer capacity. We extend their analysis to treat the finite buffer case. A key to the analysis in [Lin96] is that an evolution of the system behavior is represented by the simple birth-and-death process as shown in Figure A.1, where the total number of jobs in multiple queues is considered as a Markov state. To take account of the scheduling policy, we consider the state-dependent service rates. As shown below, the algorithm requires an iteration, and state-dependent service rates at the *i*-th iteration are represented by $\mu_k^{(i)}$ $(k = 1, 2, 3\ldots)$. Our modification for treating the finite buffer is rather straightforward, and we therefore show only the results without explanation.

By letting μ be the packet transmission rate on each wavelength, we can determine the state-dependent service rates from the following equations:

$$\mu_1^{(i+1)} = \mu \tag{4}$$

$$\mu_k^{(i+1)} = a_k(\mu_{k-1}^{(i)} + \mu) + b_k B_k, \ k = 1,\ldots,W-1 \tag{5}$$

$$\mu_k^{(i+1)} = a_k A_k + b_k B_k, \ W \le k \tag{6}$$

where

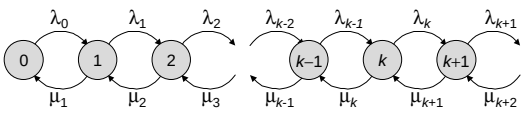

Figure A.1: State transition diagram.

$$A_k = \begin{cases} \sum_{j=0}^{\alpha_{k-1}} \binom{\alpha_{k-1}}{j} \left(q_{k-1}\right)^j \left(1-q_{k-1}\right)^{\alpha_{k-1}-j} \\ \qquad \times g_k(j+1), & \text{if } q_{k-1} < 1 \\ g_k(\alpha_{k-1}+1), & \text{if } q_{k-1} = 1 \end{cases} \tag{7}$$

$$B_k = \begin{cases} \sum_{j=0}^{\alpha_{k+1}} \binom{\alpha_{k+1}}{j} \left(q_{k+1}\right)^j \left(1-q_{k+1}\right)^{\alpha_{k+1}-j} \\ \qquad \times g_k(j+1), & \text{if } q_{k+1} < 1 \\ g_k(\alpha_{k+1}+1), & \text{if } q_{k+1} = 1 \end{cases} \tag{8}$$

$$g_k(j) = \begin{cases} (j+1)\mu, & \text{if } j < W \\ W\mu, & \text{if } j = W \end{cases} \tag{9}$$

$$h_k(j) = (j-1)\mu(1-w_k(j)) + j\mu w_k(j) \tag{10}$$

$$w_k(j) = \begin{cases} (k+1-j)/j, & \text{if } k+1-j < j \\ 1, & \text{if } k+1-j \ge j \end{cases} \tag{11}$$

$$\alpha_k = \min(k, W) - 1 \tag{12}$$

$$q_k = \frac{1}{\alpha_k}\left(\frac{\mu_k^{(i)}}{\mu} - 1\right) \tag{13}$$

$$a_k = \frac{p_{k-1}^{(i)}\lambda_{k-1}^{(i)}}{p_{k-1}^{(i)}\lambda_{k-1}^{(i)} + p_{k+1}^{(i)}\mu_{k+1}^{(i)}} \tag{14}$$

$$b_k = \frac{p_{k+1}^{(i)}\mu_{k+1}^{(i)}}{p_{k-1}^{(i)}\lambda_{k-1}^{(i)} + p_{k+1}^{(i)}\mu_{k+1}^{(i)}} \tag{15}$$

The steady-state probabilities for the birth-and-death process can be obtained in a usual way:

$$p_k^{(i)} = p_0^{(i)} \prod_{j=0}^{k-1} \frac{\lambda_j^{(i)}}{\mu_{j+1}^{(i)}}, \ k = 1,2,\ldots \tag{16}$$

$$p_0^{(i)} = \left[1 + \prod_{j=0}^{k-1} \frac{\lambda_j^{(i)}}{\mu_{j+1}^{(i)}}\right]^{-1} \tag{17}$$

One of the main differences from the original approach can be found in Eqs. (14) through (17), where packet arrival rates are also state-dependent. In state *k* the arrival rate depends on the buffer status because we should exclude the lost packets. Since the JSQ policy is used to balance the buffer occupancy among multiple queues, we assume that jobs are evenly distributed among the queues. When each of the queues contains *k* packets and packet lengths are distributed exponentially, the probability that the packet loss occurs is given by the following equation [Cal00].

$$P_l(k) = e^{-\mu_k^{(i)}(B-1)D} \sum_{j=0}^{k-1} \frac{[\mu_k^{(i)}(B-1)D]^j}{j!}, \ k > 0 \tag{18}$$

We note here that we actually need to take account of the buffer occupancy (i.e., the "unfinished work" in the

terminology of queuing theory), but for simplicity we only consider the number of jobs queued in each buffer. The newly arriving packet is lost if none of *W* queues can accept it. Thus the state-dependent packet arrival rate for the next iteration is given by the following equations:

$$\lambda_k^{(i+1)} = \begin{cases} \lambda, & k < W \\ \lambda\left\{1 - \left[P_i(\lfloor k/W \rfloor)\right]^W\right\}, & k \geq W \end{cases} \qquad (19)$$

That is, when the number of packets in the system exceeds *W*, the newly arriving packet is lost if buffer occupancies exceed the buffer size at all of the queues.

APPENDIX B: ANALYSIS OF OPTICAL DELAY LINE BUFFERS

We follow [Cal00] to analyze the optical buffering system with fiber delay lines. When all the servers are busy, the packet is stored at the shortest queue if space is available. Since the granularity of the delay line buffer is *D*, we need an additional time given by Eq. (2), during which the server is idle. Since this extra delay is introduced when the arriving packet finds no idle servers, the mean of the fictitious packet length $1/\mu_e$ is given approximately by the following equation:

$$1/\mu_e = p_0/\mu + (1-p_0)(1/\mu + D/2) \qquad (20)$$

where p_0 is the probability that all servers are idle and is given by Eq. (18) of Appendix A. By using this mean as the new value of $1/\mu$ for next iteration, we can evaluate the effect of the optical delay line buffers.

REFERENCES

[Ban00] J. Bannister, J. Touch, A. Willner, and S. Suryaputra, "How Many Wavelengths Do We Really Need? A Study of the Performance Limits of Packet over Wavelengths," *Optical Networks Magazine*, pp. 17-28, April 2000.

[Cal00] F. Callegati, "Optical Buffers for Variable Length Packets," *IEEE Commun. Letters*, Vol. 4, No. 9, pp. 292-294, Sept. 2000.

[Chl96] I. Chlmtac, et al., "CORD: Contention Resolution by Delay Line," *IEEE Journal on Selected Areas in Communications*, Vol. 14, pp. 1014-1029, 1996.

[Dan98] S.L. Danielson, P.B. Hansen, and K.E. Stubkjaer, "Wavelength Conversion in Optical Packet Switching," *Journal of Lightwave Technology*, Vol. 16, pp. 2095-2108, 1998.

[Dan00] D.O. Awduche, Y. Rekhter, J. Drake, and R. Coltun, "Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects," *IETF Internet Draft*, draft-awduche-mpls-te-optical-02.txt.

[Dav98] B. Davie, P. Doolan, and Y. Rekhter, *Switching in IP Networks - IP Switching, Tag Switching, and Related Technologies*, Morgan Kaufmann, 1998.

[Fra99] A. Franzen, H. Sotobayashi, K. Kitayama, and I. Andonovic, "Demonstration of a High Resolution Synchronizer to Facilitate Payload Recovery at an Optical Node," *IEEE Photonic Technology Letters*, Vol. 11, pp. 1671-1673, 1999.

[Ge00] A. Ge, L. Tancevski, G. Castanon, and L.S. Tamil, "WDM Fiber Delay Line Buffer Control for Optical Packet Switching," in *Proceedings of OptiComm 2000: Optical Networking and Communications*, pp. 247-256, 2000.

[Hal98] 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, 1998.

[Hun98] D.K. Hunter, W.D. Cornwell, T.H. Gilfedder, A. Franzen, and I. Andonovic, "SLOB: A Switch with Large Optical Buffers for Packet Switching," *Journal of Lightwave Technology*, Vol. 16, No. 10, pp. 1725-1736, October 1998.

[Ito00] T. Ito, K. Fukuchi, K. Sekiya, D. Ogasahara, R. Ohhira, and T. Ono, "6.4TB/s (160x40Gbit/s) WDM Transmission Experiment with 0.8bit/s/Hz Spectral Efficiency," *European Conference on Optical Communication (ECOC2000)*, PD-1.1, Munich, Sept. 2000.

[Kes98] S. Keshav and R. Sharma, "Issues and Trends in Router Design," *IEEE Commun. Magazine*, pp. 144-151, May 1998.

[Kit99] K. Kitayama and N. Wada, "Photonic IP Routing," *IEEE Photonic Technology Letters*, Vol. 11, pp. 1689-1691, 1999.

[Lin96] H.-C. Lin and C.S. Raghavendra, "An Approximate Analysis of the Join the Shortest Queue (JSQ) Policy," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 3, pp. 301-307, March 1996.

[Lin97] S. Lin and N. McKeown, "A Simulation Study of IP Switching," *ACM SIGCOMM '97*, pp. 15-24, September 1997.

[Mea00] B. Meagher, et al., "Design and Implementation of Ultra-Low Latency Optical Label Switching for Packet-Switched WDM Networks," *IEEE Journal of Lightwave Technology*, Vol. 18, pp.1978-1987, December 2000.

[Pru86] P. Prucnal, M. Santro, and T. Fan, "Spread Spectrum Fiber Optic Local Area Network using Optical Processing," *Journal on Lightwave Technology*, vol. 4, pp. 307-314, 1986.

[Qia00] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," *Journal on High Speed Networks*, Vol. 8, No. 1, pp. 69-84, 2000.

[Sal89] J. Salehi, "Code Division Multiple Access Techniques in Optical Fiber Networks - Part I: Fundamental Principles," *IEEE Transactions on Communications*, Vol. 37, pp. 824-833, 1989.

[Sob98] H. Sotobayashi and K. Kitayama, "325nm Bandwidth Supercontinuum Generation at 10Gbit/s using Dispersion-Flattened and Non-Decreasing Normal Dispersion Fibre with Pulse Compression technique," *Electron. Letters*, Vol. 34, pp. 1336-1337, 1998.

[Tan99] L. Tancevski, A. Ge, G. Castanon, and L.S. Tamil, "A New Scheduling Algorithm for Asynchronous, Variable Length IP Traffic with Void Filling," *OFC '99*, Paper ThM7, Feb. 1999.

[Tan00] 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, No. 10, pp. 2084-2093, October 2000.

[Tak00] H. Taara, et al., "Over 1000 Channel Optical Frequency Chain Generation from a Single Supercontinuum Source with 12.5GHz Channel Spacing for DWDM and Frequency Standards," *ECOC2000*, PD-3.1, Munich, Sept. 2000.

[WAN97] "WAN Packet Size Distribution," available at http://www.nlanr.net/NA/Learn/packetsizes.html, June 1997.

[Wad99] N. Wada and K. Kitayama, "10Gb/s Optical Code Division Multiplexing using 8-chip Optical Bipolar Code and Coherent Detection," *Journal on Lightwave Technology*, Vol. 17, pp. 1758-1765, 1999.

[Wad00] N. Wada and K. Kitayama, "Photonic IP Routing Using Optical Codes: 10Gbit/s Optical Packet Transfer Experiment," *2000 Optical Fiber Conference (OFC2000)*, WM51, 2000.