A Real-Time Hardware-Based Scheduler For Next-Generation Optical Burst Switches

M. T. Anan and G. M. Chaudhry
Computer Eng. Research Lab (CERL)
School of Computing and Engineering, Univ. of Missouri-Kansas City
{mtanan, chaudhryg}@umkc.edu

Abstract – Optical burst switching (OBS) is a promising technique for next-generation optical switching networks. In traditional OBS, an entire burst is discarded when all output wavelengths are engaged at the arrival instant of the burst. A critical design issue in OBS is how to reduce burst dropping probability as a result of slow scheduling reservation or resources contention. This paper proposes a hardware-based Slotted Wavelength Assignment Pipeline (SWAP) scheduler which integrates the merits of both low computational complexity and low burst dropping probability. The key idea is to maintain all scheduled data bursts and void intervals within a scheduling time window in binary vectors. Scheduling of incoming data bursts is performed using fast hardware operations without the need to search lookup tables for void intervals on output wavelengths. The design has been verified using Verilog HDL simulation models. The performance has been evaluated for 1 Tbps switch using Poisson traffic arrivals.

Index Terms – Optical Burst Switching, Channel Scheduling, Wavelength Conversion

I. INTRODUCTION

Optical burst switching (OBS) is a promising technique for next-generation optical switched networks due to its practical balance between coarse-grained circuit switching and fine-grained packet switching. OBS takes advantage of both the huge capacity in fibers for switching and the sophisticated processing capability of electronics to achieve cost reduction and leverage the technological advances in both optical and electronic worlds [1]. This makes it a viable technology for the next generation optical network.

In OBS networks, control signaling is performed out-of-band by having few channels to be dedicated to carry headers or burst control packets (BCP). Only these channels go through optical/electronic/optical (O/E/O) conversion. Data transparency is achieved by switching data bursts all-optically at burst level. This helps achieving good utilization of network resources in presence of bursty traffic [2].

Since OBS technology relies on statistical multiplexing for data bursts, contention situations occur due to scheduling reservation or transmission conflicts [2]. This leads to high burst loss probabilities. In OBS networks, a key challenge is to design an efficient scheduling algorithm for bandwidth reservation with minimum loss.

There has been a large amount of research work addressing the issue of OBS resource contention in recent years [3-6]. Techniques designed to address this issue include one or several of the following schemes: wavelength conversion, deflection routing, optical buffering and burst segmentation. These approaches show improvement on the blocking performance, but the enhancements come at the price of high complexity in terms of both hardware architectures and scheduling algorithms [7].

This paper introduces a new OBS core node architecture together with an efficient scheduling scheme. This paper is organized as follows: Section II describes our proposed approach in terms of switch design, scheduling algorithm, and performance evaluation. Finally, section III concludes the paper and discusses future work.

II. PROPOSED APPROACH

In this paper we propose a new OBS core node architecture together with a scheduling algorithm to minimize the burst loss probability and scheduling delay. The new design combines the benefits of OBS and wavelength reassignment with limited support of optical buffering.

A. Switch Design

Fig. 1 shows the design of the proposed OBS core node architecture together with the contention resolution components. The switch architecture has two input and two output fibers each fiber could have n wavelengths for data channels and c wavelengths for control channels. The switch is composed of both optical and electronic components.

The optical components of the new switch include: optical wavelength demultiplexers (ODEMUX), optical wavelength multiplexers (OMUX), splitters, couplers, tunable wavelength converters (TWC), fiber delay lines (FDLs) based on track changer (TC) architecture and the optical cross connects (OXC) based on non-blocking 2x2 optical switch elements arranged in a banyan network configuration. The electronic components are located at the switch control unit (SCU) which includes: O/E/O converters, header decoder and forwarder, scheduler, reordering buffers for control packets, switch controlling unit and control packets regenerators.

The SCU is responsible for interpretation and regeneration of BCPs, scheduling data bursts, forwarding table lookup for data bursts, collision detection and resolution. The size of the SCU is a function of the network size in order to minimize any loss of control packets due control traffic load or SCU saturation.
When the SCU receives a burst header packet, it identifies the intended destination and consults the router signaling processor to find the intended output port. If the output port is available at the time when the data burst arrives, the SCU configures the OXC to let the data burst pass through. If the port is not available, then the OXC is configured to route the data burst to the proposed contention resolution components. The control signals define how soon before the burst arrival and how long after its departure the switching elements are made available to route subsequent bursts.

For efficiency purposes, the scheduling of a BCP is processed immediately as soon as it arrives. The new generated/updated BCP is always buffered in a reordering buffer until the data burst is actually sent out. Also, it is forwarded to the next node for further reservations. Reordering buffers are used to rearrange BCPs in case they are rescheduled.

The proposed contention resolution components consist of the tunable wavelength converters and FDLs. The FDLs are based on 2x2 track changer modules to delay a congested incoming data burst to fit into an available data channel. In case of no contention, bursts will be scheduled normally and these components will not be used.

TC architecture gives the flexibility to add delay periods using minimum hardware. TC modules employ integer multiple fiber loop delays, which delay bursts for one or more burst length durations. The purpose of using track changer is to make bursts to move on their original tracks before exiting the module. Basically, they re-circulate data bursts without the need to switch them all over again. They are controlled very easily by the SCU similar to regular switching elements.

Finally, the multiplexer (MUX) inserts the control channels and the data channels and send them out to the output fiber.

B. SWAP Scheduling Algorithm

In contrast to previous research contributions, which try to solve relevant OBS scheduling problems by designing complicated hardware and control algorithms at the network core, we solve these problems by optimizing the technology itself to support an effective control algorithm.

The proposed scheduler is implemented using a simple digital logic circuit which allows it to schedule the incoming data bursts in real-time. The proposed scheduler is called Slotted Wavelength Assignment Pipeline (SWAP) scheduler. Fig. 2 shows a schematic circuit diagram for SWAP scheduler. The circuit is composed of an array of AND, OR gates, and D-Flip-Flops equivalent to the number of time slots in scheduling window. A shift register using D-Flip-Flops is used to temporarily store the binary vector for the expected arrival data burst. The design has been implemented and verified using Verilog HDL simulation models.

Fig. 1. Proposed OBS core node architecture.

Fig. 2. Schematic circuit diagram for the SWAP scheduler.
SWAP scheduler is based on maintaining a slotted scheduling window for each output wavelength channel. Binary vectors of time slots are used to describe each channel’s availability to schedule arriving data bursts at any point of time. For each time slot, a binary digit of “0” represents an available time slot (idle) while a binary digit of “1” represents an occupied time slot (busy). SWAP assumes that each time slot represents a fixed duration which corresponds to the transmission speed of the channel for sending 1500 bytes. This heuristic is based on the maximum size of an Ethernet frame. A data burst consists of an aggregation of data frames. Similarly, the arriving data bursts (specified in the burst control packets) are also described using binary vectors based on their arrival time and the size. The binary vector size should be equivalent to each one of the output scheduling channels. The output binary vectors will be populated with the expected data bursts as soon as their corresponding burst control packet arrive. Data bursts are represented using one or more time slots depending on their size.

To schedule an incoming data burst, a “bitwise AND” is performed to check if the intended output channel is available to schedule the incoming data burst at the arrival time. If the result of the “bitwise AND” is zero, the output channel is available. Therefore, a “bitwise OR” operation is performed to schedule the data burst. Otherwise, the remaining output channels will be considered sequentially. If all output channels are not available for the arriving data burst, SWAP tries to schedule the incoming data burst after delaying it on the original wavelength. A “SHIFT Left” operation is performed on the incoming data burst vector to delay it by time slots equivalent to the contending ones until the burst is either scheduled or discarded. This means that SWAP will either perform a wavelength conversion or delay for contending bursts but not both at the same time. This is due to the timing restrictions in OBS.

For each burst control packet, the scheduler tries to find the earliest time to schedule its data burst within a maintained scheduling time window (TW). TW is a maintained duration that corresponds to the minimum time gap between each control and data packet which is called offset time. By allowing the control packet to reserve the required resources for the onward transmission within the scheduling window, a minimum offset time is always maintained between the burst header and its data burst. An optical path exists only for the duration of a burst. If the scheduler was not able to schedule or delay the incoming data burst within the scheduling window, the data burst will be dropped.

The scheduling of data bursts is based on their arrival time specified in the burst control packet. The search for available time slots within the output channels is performed vertically as opposed to other algorithms where the search is performed horizontally and vertically. Therefore, SWAP does not need to maintain a table to track all scheduled and unscheduled times (i.e., voids or idles) for each data channel within scheduler’s time window. The proposed scheduling algorithm is explained in a flowchart diagram in Fig. 3.

Initially, when a BCP arrives on the control channel ($\lambda_i$), an O/E operation is performed to convert it into electronic domain. Then, the incoming data burst’s arrival time ($t_i$), the burst size ($L$), and the incoming data channels ($\lambda_i$) are obtained. The outgoing link is determined according the destination address. Then, the data burst is scheduled based on one of the following stages:

**Stage I:** Schedule arriving burst on the same wavelength $\lambda_i$ (no wavelength conversion):

\[
 b_i \leftarrow \text{arriving burst} \\
 \text{if } (b_i \& \lambda_i) = 0 \Rightarrow (\lambda_i \leftarrow b_i \mid \lambda_i) \\
 \text{else go to Stage II}
\]

**Stage II:** Schedule with other output wavelengths (using wavelength conversion):

\[
 \forall \lambda_k \mid i < k < \lambda_i \text{ : if } (b_i \& \lambda_k) = 0 \Rightarrow (\lambda_k \leftarrow b_i \mid \lambda_k) \\
 \text{where } K : \lambda_i \mod n \\
 \text{else go to Stage III}
\]

**Stage III:** Schedule with limited FDL delay:

\[
 b_i \leftarrow (b_i \& \lambda_i) \\
 \text{if } (b_i \& \lambda_i) = 0 \Rightarrow (\lambda_i \leftarrow b_i \mid \lambda_i) \\
 \text{else Discard } b_i
\]

![Fig. 3. A Flowchart diagram for SWAP scheduling algorithm.](image-url)
Fig. 4 illustrates an example of using SWAP-Stage I to schedule an incoming data burst successfully. The incoming data burst is scheduled to arrive on \((\text{fiber 1}, \lambda_2)\) at time \(t_i\) and going out on \text{fiber 1} as well. SWAP tries to schedule the incoming burst on the same output wavelength according to the wavelength continuity constraint and to avoid wavelength conversion as much as possible. Using stage I, a “bitwise AND” is performed between \(b_i\) and \(\lambda_2\). Since, the result is zero, the output channel is available. A “bitwise OR” is performed between \(b_i\) and \(\lambda_2\). The new burst was successfully scheduled on \(\lambda_2\) as shown Fig. 4 (b).

Fig. 5 illustrates stage II and III of the proposed scheduling pipeline. The incoming data burst is scheduled to arrive on \((\text{fiber 1}, \lambda_2)\) at time \(t_i\) and going out on \text{fiber 1} as well. However, the output channel \(\lambda_2\) was not available to schedule \(b_i\) due to two conflicting time slots as shown in Fig. 5 (a). The scheduler will go to Stage II and try to schedule the burst on the next channel \(\lambda_j\) on the same fiber. However, channel \(\lambda_j\) is not available due to four conflicting time slots. Therefore, the schedule uses Stage III to delay the incoming data burst on the initial output channel \(\lambda_2\) by time slots equivalent to the original conflicting slots (two time slots). Channel \(\lambda_j\) is now available to schedule the burst. In this case, FDL delay is required to schedule the incoming burst successfully.

### C. Performance Evaluation

In this section, the performance of SWAP is studied through simulation experiments. The performance metrics used are the average burst loss ratio and average burst queuing delay (or elapsed scheduling delay). The loss is calculated as the ratio of the dropped information to the total information sent ratio.

In our simulation, we consider an OBS node with two input/output fibers and 50 data wavelengths in each fiber. We assume that each group of 25 wavelengths is handled by one control channel and one scheduler. The link transmission rate is 10 Gbps. This means that our system simulates a 1 Tbps which consists of a total of 100 data wavelengths, 4 control channels, and 4 schedulers.

Our simulation studies the impact of changing offset times. This was achieved by changing the scheduling time window from 32 time slots (maximum offset time of 38.4 \(\mu s\)) to 64 time slots (maximum offset time of 76.8 \(\mu s\)). The scheduling window is represented using binary vector of time slots where each time slot represents 1500 bytes of data.

Poisson traffic using exponentially distributed arrivals was used to evaluate the performance of our system. Each burst is characterized by an arrival time, a length and the destination. The burst length is an exponentially generated random number rounded to the nearest integer multiple of the fixed sized packet length of 1500 bytes. Simulation was performed using 1 million bursts per experiment.

We assume burst header processing time of 1 \(\mu s\) and maximum FDL delay equivalent to the maximum switching time window size (either 32 time units or 64 time units). The wavelength converters have switching times of 1 \(\mu s\). Also, we assumed the following logical operations’ delay based on the TSMC 0.18\(\mu m\) technology: D-FF shift delay of 0.254 ns, OR gate delay of 0.429 ns, and AND gate delay of 0.326 ns.

Fig. 6 shows the performance SWAP scheduler in terms of average loss ratio and average scheduling delay. This simulation uses Poisson burst arrivals. As expected, the blocking probability and scheduling delay increase as load increases.

The average loss ratio and the average BCP queue waiting delay have been evaluated using SWAP scheduler over a 1 Tbps node. Fig. 7 plots the average burst loss ratio (burst loss probability) as a function of traffic load using Poisson traffic for two offset times: 38.4\(\mu s\) and 76.8 \(\mu s\). As expected, the larger the offset time, the lower the loss ratio. This is due to the higher probability of a successful transmission since the scheduler will have more time to schedule contending bursts over a larger scheduling window. However, the scheduling delay will increase. Fig. 8 clearly shows this fact.

### III. Conclusion

In this paper, we proposed a new core node architecture together with a scheduling algorithm to minimize loss rate and improve scheduling delay. The new design combines wavelength conversion capability for burst rescheduling and
limited delay to accommodate conflicting data bursts that fall within the same scheduling time window.

The proposed approach supports OBS switching for variable length bursts with low computational complexity compared to other scheduling algorithms presented in the literature. Scheduling of incoming data bursts is performed using fast digital logic operations without the need to search in lookup tables for available time slots on output wavelength channels.

Simulation results showed the performance of SWAP scheduler in terms of loss rate and scheduling delay. It is clear that SWAP does not efficiently utilize scheduling time slots due to the use of fixed time slot’s size heuristic. Arrival bursts that have smaller size than 1500 byte will occupy one time slot. However, bursts that have a slightly bigger size than 1500 byte will occupy additional time slot. This disadvantage comes as a trade off for improving the loss rate and delay.

A future enhancement to the proposed architecture will add the ability to support QoS classes via burst preemption based on priorities and traffic type. The packet scheduler algorithm shall consider issues such as fairness and QoS classes.

REFERENCES