Optical Burst Switching (OBS): A New Area in Optical Networking Research

Yang Chen, Chunming Qiao and Xiang Yu
Computer Science and Engineering Department
State University of New York at Buffalo
{yangchen,qiao,xiangyu}@cse.buffalo.edu

Abstract

In this tutorial, we give an introduction to optical burst switching (OBS) and compare it with other existing optical switching paradigms. Basic burst assembly algorithms and their effect on assembled burst traffic characteristics are described first. Then a brief review of the early work on burst transmission is provided followed by the description of a prevailing protocol for OBS networks called Just-Enough-Time (JET). Algorithms used at an OBS core node for burst scheduling as well as contention resolution strategies are presented next. Tradeoffs between their performance and implementation complexities are discussed. Recent work on QoS support, IP/WDM multicast, TCP performance in OBS networks and Labelled OBS is also described, and several open issues are mentioned.

1 Introduction

With recent advances in wavelength division multiplexing (WDM) technology, the amount of raw bandwidth available in fiber links has increased by many orders of magnitude. Meanwhile, the rapid growth of Internet traffic requires high transmission rates beyond a conventional electronic router's capability. Harnessing the huge bandwidth in optical fiber cost-effectively is essential for the development of the next generation optical Internet.

Several approaches have been proposed to take advantage of optical communications and in particular optical switching. One such approach is optical circuit switching based on wavelength (λ) routing whereby a lightpath needs to be established using a dedicated wavelength on each link from source to destination. Once the connection is set up, data remains in the optical domain throughout the lightpath. An alternative to optical circuit switching is optical packet switching. In optical packet switching, while the packet header is being processed either all-optically or electronically after an Optical/Electronic (O/E) conversion at each intermediate node, the data payload must wait in the fiber delay lines and be forwarded later to the next node [1, 2].

In order to provide optical switching for next generation Internet traffic in a flexible yet feasible way, a new switching paradigm called optical burst switching (OBS) was proposed in [3–5]. Various OBS approaches with different tradeoffs have since been described (see papers listed in http://www.cse.buffalo.edu/∼yangchen/OBS_Pub_year.html, http://www.utdallas.edu/∼vinod/obs.html and http://www.ikr.uni-stuttgart.de/ ∼gauger/BurstSwitching/). There are two common characteristics among these variants:

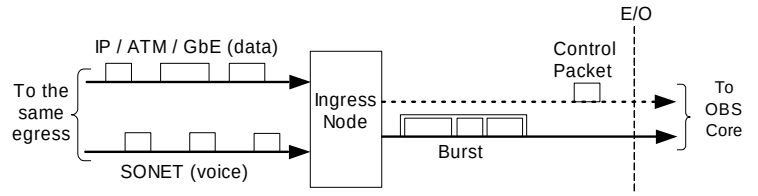
- Client data (e.g., IP packets) goes through burst assembly/disassembly (only) at the edge of an OBS network, nevertheless, statistical multiplexing at the burst level can still be achieved in the core of the OBS network.
- Data and control signals are transmitted separately on different channels or wavelengths (λ's)[1], thus, costly O/E/O conversions are only required on a few control channels instead of a large number of data channels.

In this tutorial, we first introduce the basic idea of OBS, compare it with other switching paradigms and point out why OBS is a viable technology for the next generation optical Internet. Techniques for generating a burst at the edge

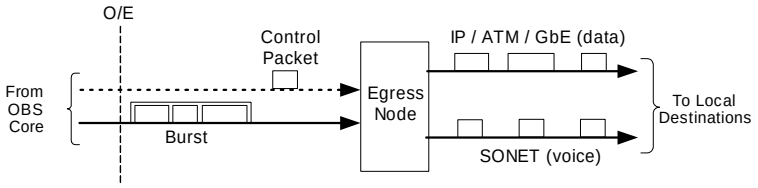
[1]Hereafter, we will use the terms channel and wavelength (or λ) interchangeably.

of an OBS network are studied next, followed by discussions on various burst reservation protocols. Characteristics of the burst traffic assembled using different assembly algorithms are also analyzed. The subsequent section focuses on issues at a core OBS node: namely burst scheduling and contention resolution. Recent efforts on supporting service differentiation, IP/WDM multicast, the performance of TCP and other issues related to traffic engineering in OBS networks are described at the end.

2 OBS Fundamentals



(a) Burst Assembly



(b) Burst Disassembly

Figure 1: Burst Assembly/Disassembly at the Edge of an OBS Network

In an OBS network, various types of client data are aggregated at the ingress (an edge node) and transmitted as data bursts (Figure 1(a)) which later will be disassembled at the egress node (Figure 1(b)). During burst assembly/disassembly, the client data is buffered at the edge where electronic RAM is cheap and abundant.

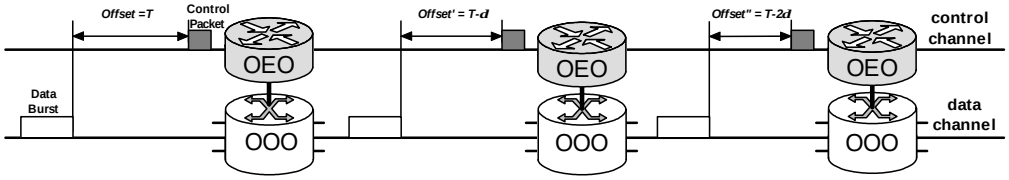
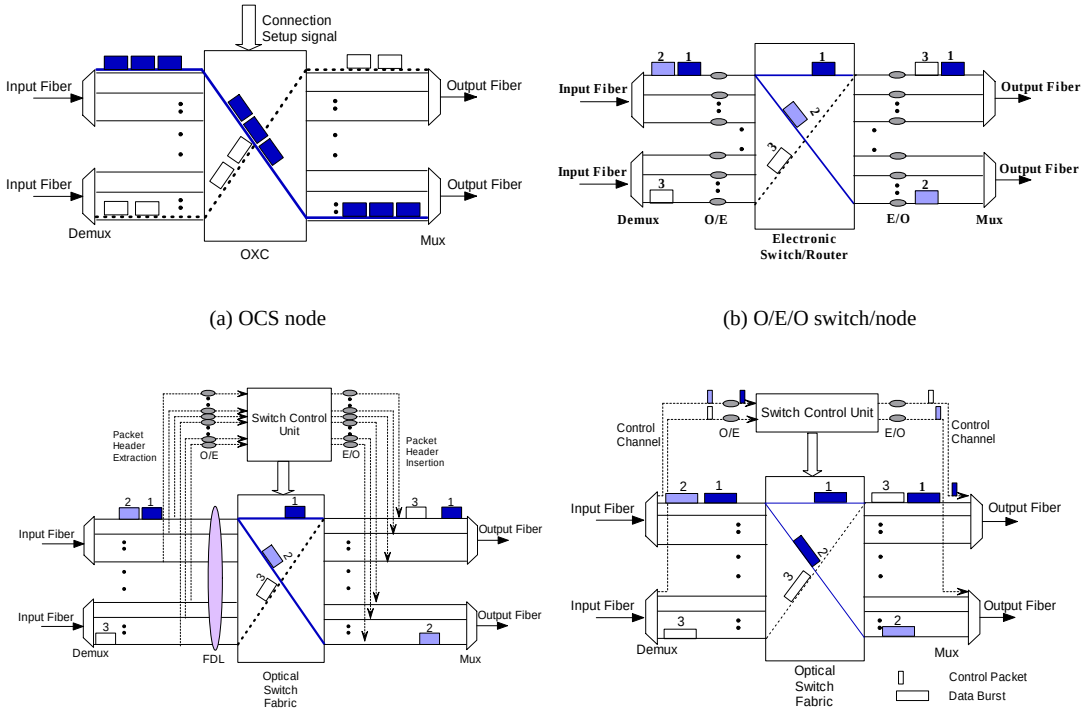


Figure 2: Separated Transmission of Data and Control Signals

Figure 2 depicts the separation of data and control signals within the core of an OBS network. For each data burst, a control packet containing the usual ”header” information of a packet including the burst length information is transmitted on a dedicated control channel. Since a control packet is significantly smaller than a burst, one control channel is sufficient to carry control packets associated with multiple (e.g., hundreds of) data channels. A control packet goes through O/E/O conversion at each intermediate OBS node and is processed electronically to configure the underlying switching fabric. There is an offset time between a control packet and the corresponding data burst to compensate for

the processing/configuration delay. If the offset time is large enough, the data burst will be switched *all-optically* and in a "cut-through" manner, i.e., without being delayed at any intermediate node(core). In this way, no optical RAM or fiber delay lines (FDLs) is necessary at any intermediate node. Nevertheless, the burst-level granularity leads to a statistical multiplexing gain which is absent in optical circuit switching. Furthermore, it allows a lower control overhead per bit than that in optical packet switching as to be discussed next in more detail.

3 Qualitative Comparison



(a) OCS node

(b) O/E/O switch/node

(c) OPS node

(d) OBS node

Figure 3: Comparison of Different Switching Node Architectures

Basic switching node architectures used by various switching paradigms are illustrated and compared in Figure 3. At an optical circuit switching (OCS) node shown in Figure 3(a), once a lightpath is setup, all data carried by one input λ will go to a specific output λ. Since no O/E/O conversion of data at any intermediate node is needed, multi-hop transparency (in terms of the bit rate, protocol and coding format used) can be achieved. On average, the connection duration should be on the order of minutes or longer as setting up or releasing a connection takes at least a few hundreds of milliseconds. Shorter duration connections needed to accommodate sporadic data transmissions will result in a prohibitively high control overhead. A major difference between OCS and the other three approaches depicted in Figure 3 is that in OCS, no statistical multiplexing of the client data can be achieved at any intermediate node. More specifically, in the core, bandwidth is allocated by one λ at a time, which is a coarse granularity. In practice, however, most of today's applications only need the sub-λ connectivity. In addition, high-bit rate computer communications often involve "bursts" that last only a few seconds or less.

To overcome the above deficiency of the OCS approach, O/E/O conversion can be introduced above an OCS network in the IP and SONET layers for example. The electronic switching node used in such an O/E/O approach is depicted in Figure 3(b). Here, statistical multiplexing of the client data at the sub-λ granularity is possible with electronic

processing and buffering (not shown in Figure 3(b)). Since every data unit needs to go through O/E and E/O conversion, this approach is not scalable enough to support hundreds of wavelengths, each working at 40Gbps or beyond (the need for which is anticipated in the near future). In addition, electronic switches are known to suffer from problems such as limited capacity and huge power/space consumption and heat dissipation in addition to requiring expensive O/E/O conversions. Note that, although not shown, either an optical cross connect or optical add-drop multiplexor may also be used in conjunction with an electronic switch for wavelength granularity traffic that does not need to go through the electronic switch. A hybrid, multi-layer network consisting of such nodes, each consisting of both an electronic switch/router and an optical cross connect, is one way to combine the strength of the optics and electronics, but certainly not the only way to do so, and in fact may not be the ultimate long-term solution.

Since all-optical header processing will not be economically viable in the near future due to the immaturity of high-speed optical logic, the optical packet switching (OPS) approach will likely require each header to go through O/E conversion for processing and E/O conversion for transmission (Figure 3(c)). An important difference from the previous O/E/O approach is that here, the header can potentially be sent at a much slower rate than the data using for instance sub-carrier multiplexing, thereby easing the speed requirement on the O/E/O conversion devices while still maintaining a high data throughput. Nevertheless, OPS is difficult to implement because of its need for a large number of O/E/O conversion devices (one set for each wavelength), header extraction/insertion mechanisms (though not shown in the figure) as well as FDLs and packet synchronizers. Note that, an optical cross-connect or add-drop multiplexor mentioned above can also be used in conjunction with the OPS nodes or OBS nodes to be discussed below if/when it is more economic to do so.

In the OBS paradigm, only a few control channels (e.g., one per fiber) go through O/E/O conversion (see Figure 3(d)). Given that the data is switched all-optically at burst level, data transparency and statistical multiplexing can be achieved concurrently. Since OBS takes advantage of both the huge capacity in fibers for switching/transmission and the sophisticated processing capability of electronics, it is able to achieve cost reduction and leverage the technological advances in both optical and electronic worlds, which makes it a viable technology for the next generation optical Internet.

At an OBS node, no synchronization/alignment of bursts is necessary unless the switching fabric operates in a slotted manner. In addition, FDLs and wavelength converters which are optional can help in reducing burst loss [6]. Currently, it is a challenge to implement an OBS switching fabric with hundreds of ports operating at a switching speed which is on the order of nanoseconds. Nevertheless, on-going research work has shown promise [7–9].

4 Burst Assembly

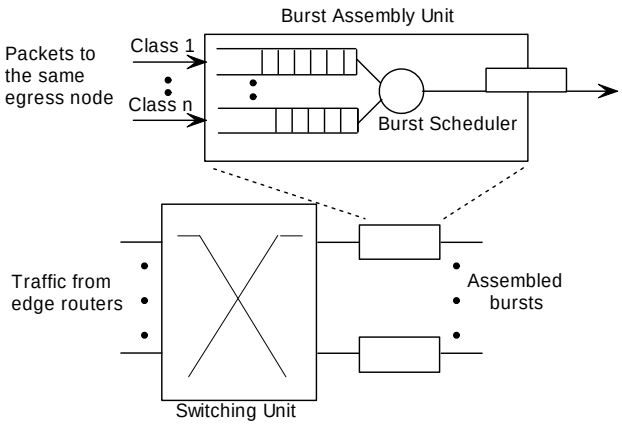


Figure 4: Architecture of an OBS Ingress Node

Burst assembly is the procedure of aggregating packets from various sources, such as an IP router, into bursts at

the edge of an OBS network. The architecture of a typical OBS ingress node is shown in Figure 4. The switching unit forwards incoming packets to burst assembly units. The packets to the same OBS egress node are processed in one burst assembly unit. Usually, there is one designated assembly queue for each traffic class (or priority). The burst scheduler is in charge of creating bursts and their corresponding control packets, adjusting the offset time for each burst, scheduling bursts on each output link and forwarding the bursts and their control packets to the OBS core network [7].

Recent studies on burst assembly have shown that different assembly schemes affect the assembled burst traffic's characteristics [10,11]. In the following subsections, we will describe several assembly algorithms and discuss statistical characteristics of the assembled burst traffic.

4.1 Assembly algorithms

Usually, assembly algorithms can be classified as *timer-based*, *burstlength-based* and *mixed timer/burstlength-based* ones [11, 12].

In the *timer-based* scheme, a timer starts at the beginning of each new assembly cycle. After a fixed time T, all the packets that arrived in this period are assembled into a burst. In the *burstlength-based* scheme, there is a threshold on the (minimum) burst length. A burst is assembled when a new packet arrives making the total length of current buffered packets exceed the threshold.

The time out value for *timer-based* schemes should be set carefully. If the value is too large, the packet delay at the edge might be intolerable. If the value is too small, too many small bursts will be generated resulting in a higher control overhead. While *timer-based* schemes might result in undesirable burst lengths, *burstlength-based* assembly algorithms do not provide any guarantee on the assembly delay that packets will experience. To address the deficiency associated with each type of the assembly algorithms mentioned above, *mixed timer/threshold-based* assembly algorithms were proposed in [7, 11]. For example, in [11], a burst can be sent out when either the burst length exceeds the desirable threshold or the timer expires.

Adaptive assembly algorithms were also proposed to optimize the performance of OBS networks in which either the time threshold or the burst length threshold or both are adjusted dynamically according to real time traffic measurements. They provide better performance especially with strongly correlated input packet traffic but have a higher operational complexity.

After a burst is generated using the algorithms mentioned above, the burst is buffered in the queue for an offset time before being transmitted to give its corresponding control packet enough time to make reservations at the downstream nodes as shown in Figure 2. During this offset period, packets may continue to arrive. Including those packets in the same burst is usually unacceptable because the reservation at the downstreams nodes may have already been made based on the original burst length record in the control packet. Leaving those packets for the next burst on the other hand, will increase the average delay especially when the traffic load is heavy. One way to minimize this extra delay is to perform burst length prediction: let the control packet carry a burst length of $l + f(t)$ instead of l, where l is the exact burst length when the control packet is sent out, and $f(t)$ is the *predicted* extra burst length as a result of additional packet arrivals during the offset time t. Assume that the total length of packets actually arriving during the offset time is $l(t)$. If $l(t) < f(t)$, part of bandwidth reserved will be wasted. Otherwise (i.e., if $l(t) > f(t)$), only a few extra packets (whose total length is about $l(t) - f(t)$) are delayed to be transmitted in the next burst.

4.2 Assembled burst traffic characteristics

Recently, the sensitivity of OBS network performance to the assembled burst traffic characteristics such as inter-arrival time and burst length distribution has been studied. these studies have focused on the statistical characteristics of burst traffic, which can be divided into two categories: short range (small time scales) and long range (large to infinite time scales). In most of these studies, the packet arrivals into an assembly queue from many independent traffic sources were assumed to be Poisson. For a *timer-based* assembly algorithm, the size of a burst is equal to the sum of the size of all the packets arriving in a fixed time period, and was shown in [11] to be a Gaussian distributed random variable according to the central limit theorem. Conversely, for a *burstlength-based* assembly algorithm, the burst inter-arrival times have a Gaussian distribution. Similar observations were reported via simulation in [7,10,13]. The general conclusion is that the

short range burstiness in the input packet traffic is alleviated due to burst assembly and the smoothed assembled burst traffic can enhance the network's performance.

An important characteristic of today's Internet traffic is its long range dependency, which increases data loss and delay and decreases network resource utilization in electronic packet switched networks. Although it was claimed in [12] that burst assembly algorithms could reduce the long range dependency in the input IP packet traffic, [7, 11] pointed out that long range dependency in the traffic will not change after burst assembly. On the other hand, the results in [14] showed that the influence of the long range dependency on the performance of an OBS node (i.e., in terms of burst loss rate) is negligible because of its bufferless nature.

If a *timer-based* assembly scheme is used, the bursts' inter-arrival time will be a constant. Furthermore, if a *burstlength-based* assembly algorithm is used, the variance of the inter-arrival time of the bursts coming from different edge nodes may become small when the traffic load is heavy. In such cases, undesirable persistent collisions of bursts from different sources might happen if these sources are adversely synchronized. Adding a randomized extra offset time at each edge node may prevent such synchronization among the sources.

5 Burst Reservation Protocols

Although the concept of burst switching was introduced for centralized TDMA systems [15] and ATM networks [16] in early 1990, protocols suitable for high speed WDM optical networks were not developed until 1997 [3]. In this section, we will first give an overview of early burst transmission protocols followed by an introduction to protocols for OBS networks.

In [16], the author evaluated two burst level admission control mechanisms for ATM networks: tell-and-wait and tell-and-go. In the former, when a source has a burst to transmit, it first tries to reserve the bandwidth/wavelength from the source to its destination by sending a short 'request' message. Every intermediate node receiving this message will make a reservation on a specific output link. If the requested bandwidth is successfully reserved on all the links along the path, an ACK will be sent back to inform the source to send out the burst immediately; Otherwise, a NAK will be returned to release the previously reserved bandwidth, and initiate the retransmission of the 'request' message after a backoff time. In tell-and-go, on the other hand, the source transmits bursts without making any bandwidth reservation in advance. At an intermediate node, the burst needs to be delayed before the switch control unit makes an appropriate reservation on an outgoing link. If the reservation fails at any intermediate node, a NAK will be sent back to the source to initiate the retransmission of the burst after a backoff time.

Various performance comparisons (in terms of e.g., throughput and delay) between these two conceptual approaches were given in [16]. It has been found that tell-and-wait outperforms tell-and-go when the propagation delay is negligible with respect to the burst length. The opposite becomes true when the propagation delay is significant compared with the burst length.

The concept of tell-and-go forms the basis of Terabit Burst Switching [5]. With this approach, in order to compensate for the control packet processing time and prevent a burst from entering the switching fabric before its configuration is finished, a fixed delay is inserted into the data path using a FDL at each input port. On the other hand, Just-In-Time (JIT), which was first proposed in [15], can be considered as a variant of tell-and-wait as it requires each burst transmission request to be sent to a central scheduler. The scheduler then informs each requesting node the exact time to transmit the data burst. Here, the term *Just-In-Time* means that by the time a burst arrives an intermediate node, the switching fabric has already been configured. This concept was later applied and extended to a Wavelength Routed OBS network [13]. Since centralized protocols are neither scalable nor robust, [17] provided a distributed version of JIT protocol called Reservation with just-In-Time, which requires a copy of the request to be sent to all switches (each has a scheduler) concurrently. These schedulers are not only synchronized in time, but also share the same global link status information, which makes the implementation difficult. The authors of [18] proposed another distributed version of the JIT protocol based on hop-by-hop reservation which adopts some features of the Just Enough Time (JET) protocol [3, 4].

JET is the most prevailing distributed protocol for OBS networks today which does not require any kind of optical buffering or delay at each intermediate node [3, 4]. It accomplishes this by letting each control packet carry the offset time information and make the so called delayed reservation for the corresponding burst, i.e., the reservation starts at the expected arrival time of the burst. In the example shown in Figure 5, the bandwidth is reserved for the first burst starting

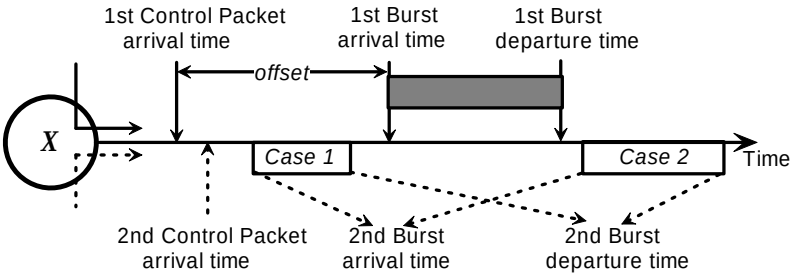


Figure 5: JET Protocol

from the burst arrival time instead of the arrival time of control packet. At each intermediate node, the offset time is updated (reduced) to compensate for the actual control packet processing/switch configuration time (see Figure 2). Note that the delay experienced by a control packet might vary for different reasons. In addition, when we consider deflection routing in an OBS network, the minimal offset time for the primary path might not be enough if the burst takes a longer alternate path. In such a case, an extra offset time can be added [4].

Another important feature of JET is that the burst length information is also carried by the control packet, which enables it to make a closed-ended reservation (i.e., only for the burst duration with automatic release) instead of an open-ended reservation (i.e., which would not be terminated until a release signal is detected). This closed-ended reservation helps the intermediate node make intelligent decisions as to whether it is possible to make a reservation for a new burst and thus the effective bandwidth utilization can be increased. An example is shown in Figure 5 where the reservation for the 2nd burst arrival in Cases 1 and 2 can succeed if and only if at the time when the 2nd control packet arrives, the intermediate node makes closed-ended reservations for both the first and second bursts.

6 Burst Switching

In a conventional electronic router/switch, contention between packets can be resolved by buffering. However, in OBS networks, no or limited buffering is available and thus burst scheduling and contention resolution must be done in a different manner.

6.1 Scheduling Algorithms

When wavelength conversion capability is assumed, an incoming burst may be scheduled onto multiple wavelengths at the desired output port. A burst scheduler will choose a proper wavelength for this burst taking into consideration the existing reservations made on each wavelength, and make a new reservation on this selected channel. Below, we will describe several scheduling algorithms.

The scheduling horizon is defined as the latest time at which the wavelength is currently scheduled to be in use. In Figure 6, for example, time t_1'' is the scheduling horizon for channel C_1. A simple scheduling algorithm: Horizon [5], which is also called the LAUC (latest available unscheduled channel) algorithm in [7] works as follows, for each wavelength, a single *scheduling horizon* is maintained. Only the channels whose scheduling horizons precede the new burst's arrival time are considered "available" and the one with the latest scheduling horizon is chosen. The horizon is then updated after making the reservation for the next burst. The basic idea for this algorithm is to minimize bandwidth gaps/voids created as a result of making a new reservation. In Figure 6, channel C_3 will be reserved if Horizon is applied.

Simplicity in both operation and implementation is the main advantage of the Horizon-based algorithms. However, they waste the gaps/voids between two existing reservations, e.g., $t_1' - t_1$ on channel C_1 in Figure 6. When a FDL set is available or the offset-time based QoS [19] scheme to be mentioned in the following section is applied, many such voids will be generated. Therefore, algorithms capable of void filling, i.e., making new reservations within existing gaps are

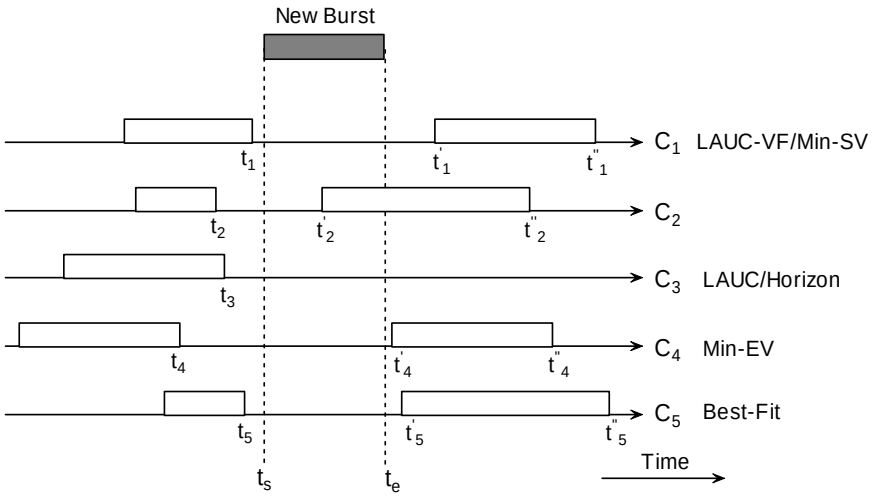


Figure 6: Illustration of Scheduling Algorithms

desirable. For example, using LAUC-VF (LAUC with void filling) proposed in [7], channel C_1 will be chosen.

Several variants of the LAUC-VF algorithm including Min-SV (Starting Void), Min-EV(Ending Void) and Best Fit were proposed in [20]. Min-SV is functionally the same as LAUC-VF but a much faster implementation is achieved using a technique from computational geometry. On the other hand, Min-EV tries to minimize the new void generated between the end of new reservation and an existing reservation while Best Fit tries to minimize the total length of starting and ending voids generated after the reservation. Figure 6 illustrates the outcomes of these three scheduling algorithms.

Algorithms	Time complexity	State information	Bandwidth Utilization
LAUC	$O(W)$	$Horizon_i$	Low
LAUC-VF	$O(WlogM)$	$S_{i,j}$, $E_{i,j}$	High
Min-SV/EV	$O(logM)$	$S_{i,j}$, $E_{i,j}$	High
Best-Fit	$O(log^2M)$	$S_{i,j}$, $E_{i,j}$	High

Table 1: Comparison of Different scheduling Algorithms

The performance of various scheduling algorithms was compared in [20], which shows that LAUC-VF, Min-SV, Min-EV and Best Fit have a comparable bandwidth utilization (or loss rate) which is much higher (or lower) than Horizon based algorithms. The running time complexity of different scheduling algorithms was also analyzed. Table 1 summarizes the above discussion using the following notations:

- W: Number of wavelengths at each output port
- M: Maximum number of data bursts (or reservations) on all channels
- $Horizon_i$: Horizon of the i^{th} data channel
- $S_{i,j}$ and $E_{i,j}$: Starting and ending time of j^{th} reservation on channel i

From Table 1, the Min-SV/EV algorithms are the most desirable among all void-filling algorithms. In fact, one can minimize the void newly generated by first searching for a proper void using Min-EV first, and then if (and only if) such a proper void cannot be found, search for a horizon using Min-SV.

6.2 Contention Resolution

Using one way reservation protocols such as JET, the ingress node sends out bursts without having reservation acknowledgements or global coordination. This however, requires an intermediate OBS node to resolve possible contention among bursts. In a bufferless OBS network, contention among the bursts can be resolved in three ways: deflection, dropping and preemption.

Through deflection, a burst is sent to a different output channel instead of the preferred one. Since contention can only happen when bursts compete for the same wavelength on the same output port simultaneously, deflection can be applied in wavelength, space and/or time domains.

- Wavelength domain: a contending burst can be sent on another wavelength through wavelength conversion.
- Space domain: a contending burst can be sent to a different output port and then follow an alternate route to the destination [4].
- Time domain: by passing through an FDL, a contending burst can be delayed for a fixed time [4, 19].

If a contending burst cannot be deflected due to the unavailability of any wavelength, output port or FDL, data loss becomes inevitable. More specifically, a common approach is to drop the incoming burst (which is a non-preemptive approach). In addition, it is possible for the incoming burst to preempt an existing burst based on priority or traffic profile. It is also possible to break the incoming burst or the existing burst into multiple segments, and each segment can then be deflected, dropped or preempted. This approach was called burst segmentation in [21, 22] and OCBS in [23].

Contention resolution	Advantages	Disadvantages
Wavelength Conversion	Much lower burst loss	Immature and expensive technology
FDL buffer	Conceptually simple; Mature technology	Bulky FDLs; Extra delay; More voids
Deflection routing	No extra hardware requirement	Out of order arrivals; Possible instability
Burst Segmentation	Finer contention resolution	Complicated control

Table 2: Comparison of Different Contention Resolution Schemes

Table 2 gives a brief summary of these contention resolution schemes. Note that some of these contention resolution schemes can be applied jointly. For example, instead of simply forwarding a burst onto an alternate route (using deflection routing) when contention happens, one can deflect a burst along a pre-determined path that returns the burst to the node where the deflection occurred and then forwards it along the original route. With this approach, the network acts like a buffer (or FDL).

Unlike all the contention resolution schemes mentioned above which work in a passive manner, i.e., taking certain actions after a contention occurs, one may collect the burst loss performance statistics on different wavelengths and rank them with priorities accordingly. Bursts are then assigned to higher priority wavelengths which have lower burst loss rates whenever possible [24]. However, this approach can only be taken by edge nodes in a network without wavelength converters. We observe that one can also pro-actively reduce burst contention (and loss) by using either the electronic buffer at an edge node or FDLs at an upstream node to sequentialize the bursts on as few wavelengths as possible so as to reduce the number of bursts which might cause overlapped reservations on an output channel at a downstream node.

7 Towards an Optical Internet

Over the past few years, running IP applications directly above the optical layer has received a considerable amount of attention. In this section, we give an overview of several initial steps toward building an optical Internet based on the OBS paradigm.

7.1 Service differentiation

Much research work has been devoted to QoS provisioning in the Internet. However, various Internet service disciplines and packet scheduling algorithms developed in the literature are based on electronic packet switching and mandate the use of buffers. In the optical domain, a FDL can provide a limited and deterministic delay but it is incapable of providing most of the buffer management functions as an electronic RAM does. To address the discrepancy between a bufferless OBS network and an electronic packet switched network, three different approaches have been proposed to provide service differentiation in OBS networks.

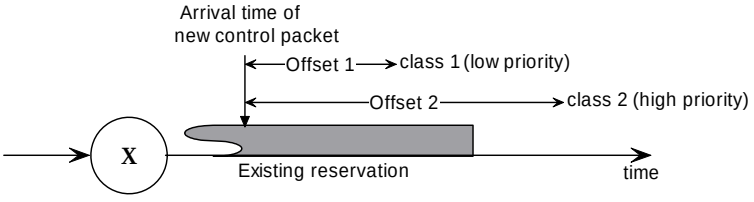


Figure 7: Offset Time's Effect on Burst Loss

The first approach manages QoS on a class by class basis using different *extra* offset times for different classes of bursts [19]. The basic idea is by giving a larger extra offset time to a higher priority class, reservation for a higher priority burst can be made in much advance than lower priority bursts and thus has a better chance to succeed. Figure 7 shows that a long offset time enables a high priority class burst to succeed in making a reservation. Studies have shown that the probability that a low priority burst will block a high priority burst can be negligible when the difference of offset time between these classes is a few times of the average burst length of the low priority class [19].

Although the offset time based differentiation is easy to implement and provides efficient isolation between service classes when a sufficiently large extra offset time is assigned to higher priority bursts, the extra offset time introduces an additional delay at the edge and in addition, the performance of the differentiation depends on the burst length and inter-arrival time distributions. Active dropping was thus proposed to avoid the shortcomings mentioned above [25, 26]. In this alternative to the offset time based differentiation, selective dropping of bursts is initiated according to either loss rate measurement or traffic profile to guarantee that the high priority class will have a better chance to make successful reservations.

While the above two approaches can provide differentiation at the burst level, differentiation at the packet level can be achieved with burst segmentation. In such an approach, packets from different service classes are assembled into different bursts. When contention occurs, low priority bursts will be segmented and experience a higher packet loss probability. Alternatively, instead of assembling a burst with packets from a single service class only, packets from low priority service classes can be assembled to form the tail or head of each burst, whereas packets from high priority service classes are assembled in the middle of each burst. If segments at the tail or head of a burst are dropped when contentions happen, differentiation on packet loss can be achieved [22].

7.2 IP/WDM Multicast and TCP over OBS

Many of today's and emerging Internet applications can be more efficiently supported using multicast. A straightforward way to do multicasting in an OBS network is Separate Multicast in which multicast traffic and unicast traffic are assembled separately into different bursts. To reduce the overhead due to guard bands and control packet associated with each burst, a scheme called Tree-Shared Multicasting was proposed [27], whereby multicast traffic belonging to different multicast sessions can be assembled together in a burst, which is then delivered via a shared multicast tree. Various criteria for determining whether two multicast sessions should share a tree and various algorithms for constructing the shared multicast trees were presented in [27]. Since it is possible that some data in a burst is delivered to non-intended destinations via a shared multicast tree, the benefit of multicast sharing strategy depends on the degree of overlapping among the multicast sessions that share the multicast tree.

OBS networks also have unique characteristics that affect TCP throughput performance. Since TCP is the most

widely used protocol for data transmissions, understanding TCP performance in an OBS network, which may become a future Internet backbone and support a large amount of TCP traffic, is thus of much interest. Several recent studies have investigated the interactions between OBS and TCP congestion control mechanisms. For example, the study in [28] found that the burst assembly process introduces a delay penalty in TCP throughput because it increases the round trip time, which in turn decreases TCP throughput. On the other hand, the enlarged transmission unit from a packet to a burst can increase the amount of data sent between two losses, resulting in the so-called "correlation gain". More specifically, a TCP source with a relative low access bandwidth in the local IP access network and a small burst assembly time at the edge can have only one TCP segment in each burst, and thus there is no correlation gain. But since the delay penalty is also insignificant in this case, the throughput is similar to that without burst assembly. For a TCP source with a relatively high access bandwidth and a large burst assembly time, all TCP segments from one sending window can be assembled into one burst, and hence, the correlation gain is maximized but the delay penalty is also large. In our recent studies, we have found that for a TCP source which has a medium access bandwidth (between low and high relative to the burst assembly time), using an adaptive assembly algorithm yields the best throughput because it can adjust the assembly time to match the TCP congestion control mechanisms.

7.3 LOBS

The generality of the evolving G-MPLS framework makes it a versatile framework for various underlying switching paradigms. For example, when G-MPLS is applied to OCS in the forms of MPλS, a wavelength is treated as the label. But such a λ- Labelled Switch Path(LSP), which corresponds to a lightpath, cannot be aggregated at the intermediate node due to the lack of wavelength merging technology. In order to groom or aggregate traffic carried on different lightpaths, each lightpath needs to go through O/E/O conversion.

As a natural extension of the G-MPLS in OBS networks, Labelled OBS(LOBS) was proposed in [29]. LOBS is built upon OBS by letting each control packet carry additional label information. One of the major benefits of LOBS is to facilitate the seamless integration of IP and WDM by using IP-based protocols for control while switching data all-optically. And unlike MPλS, the association between a label and a wavelength in LOBS is not on the time scale of a connection but that of each burst, thus making sub-wavelength granularity and statistical multiplexing possible. Even without wavelength conversion capability, bursts belonging to the same LSP (called LOBS path) can be sent on different output wavelengths at the ingress node (with a tunable transmitter). More important, the bursts belonging to different LSPs can interleave on the same wavelength, that is, bursts arriving on different LSPs (on the same wavelength) can now be merged into an aggregated LSP.

8 Concluding Remarks

In this tutorial, we have first given an introduction to optical burst switching (OBS). Comparison between this new switching paradigm with other existing optical switching paradigms has been made, and it has been shown that OBS is not only a cost-effective but also a viable solution for the next generation optical Internet. We have provided a brief historical review of the early work on burst switching as well as the state of the art including the prevailing reservation protocol for OBS networks, called Just-Enough-Time (JET) and described its major features and benefits.

This tutorial has also attempted to provide a comprehensive coverage of research issues related to OBS. Among the issues covered are various burst assembly algorithms used at the edge of an OBS network as well as their effect on traffic characteristics of the assembled burst traffic and in turn the TCP performance. We have also presented various scheduling algorithms as well as burst contention resolution strategies used in the OBS core. It has been pointed out that bandwidth-efficient scheduling algorithms like Min-SV can have a fast implementation, and that burst loss can be reduced using pro-active burst contention resolution algorithms running at the edge (in addition to the core). Finally, recent work on QoS support, IP/WDM multicast and G-MPLS extension have been discussed.

In addition to the challenges in implementing fast and scalable switching fabrics, and related devices such as FDLs and all-optical wavelength converters, there are many open issues in OBS architecture research. Chief among them are LOBS path provisioning, protection/restoration schemes as well as differentiation schemes to combat burst losses due to the inevitable burst contention in a bufferless OBS network. Others include efficient support of a mixed set

of periodical connections (to emulate SONET) and non-periodical burst transmissions, accurate single-node as well as end-to-end performance analysis in OBS networks, quantitative cost and performance comparison between OBS and other switching paradigms, and the design and evaluation of various TCP implementations over OBS.

OBS has received a lot of attention during the past few years and is fast becoming an important area of research. This tutorial will hopefully become a useful resource for researchers working on OBS or those new to this topic.

Acknowledgment

This work was supported in part by an NSF grant under contract number ANIR-9801778. The authors would like to thank the anonymous reviewers and the Editor-in-Chief for their valuable comments that have helped us improve the manuscript.

References

[1] D. J. Blumenthal, P. R. Prucnal, and J. R. Sauer, "Photonic packet switches: architectures and experimental implementations," *Proceedings of the IEEE*, vol. 82, pp. 1650–1667, November 1994.

[2] G.-K. Chang, G. Ellinas, B. Meagher, W. Xin, S.J. Yoo, M.Z. Iqbal, W. Way, J. Young, H. Dai, Y.J. Chen, C.D. Lee, X. Yang, A. Chowdhury, and S. Chen, "Low Latency Packet Forwarding in IP over WDM Networks Using Optical Label Switching Techniques," in *IEEE LEOS 1999 Annual Meeting*, 1999, pp. 17–18.

[3] M. Yoo and C. Qiao, "Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks," in *Proceeding of IEEE/LEOS Conf. on Technologies For a Global Information Infrastructure*, August 1997, pp. 26–27.

[4] C. Qiao and M. Yoo, "Optical burst switching (OBS)–a new paradigm for an optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.

[5] J. Turner, "Terabit burst switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3–16, 1999.

[6] C. Gauger, "Contention resolution in Optical Burst Switching networks," in *Advanced Infrastructures for Photonic Networks: WG 2 Intermediate Report*, 2002, pp. 62–82.

[7] Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1838–1851, October 2000.

[8] F. Masetti et.al., "Design and Implementation of a Multi-Terabit Optical Burst/Packet Router prototype," in *OFC Postdeadline Papers*, 2002, pp. FD11–FD13.

[9] J. Ramamirtham and J. Turner, "Design of Wavelength Converting Switches for Optical Burst Switching," in *Proceedings of INFOCOMM*, 2002, vol. 1, pp. 362–370.

[10] K. Laevens, "Traffic characteristics inside optical burst switched networks," in *Proceeding of Opticomm*, 2002, pp. 137–148.

[11] X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks," in *Proceeding of Opticomm*, 2002, pp. 149–159.

[12] A. Ge, F. Callegati, and L. Tamil, "On optical burst switching and Self-similar traffic," *IEEE Communications Letters*, vol. 4, pp. 98–100, March 2000.

[13] M. Düser and P. Bayvel, "Analysis of a dynamically wavelength-routed optical burst switched network architecture," *IEEE/OSA Journal of Lightwave Technology*, vol. 20, pp. 574–585, April 2002.

[14] M. Izal and J. Aracil, "On the Influence of Self-similarity on Optical Burst Switching Traffic," in *Proceeding of GLOBECOM*, 2002, vol. 3, pp. 2308–2312.

[15] D. L. Mills, C. G. Boncelet, J. G. Elias, P. A. Schragger, and A. W. Jackson, "Highball: a high speed, reserved-access, wide-area network," Tech. Rep. 90-9-3, Electrical Engineering Department,University of Delaware, 1990.

[16] I. Widjaja, "Performance analysis of burst admission-control protocols," *IEE Proceeding of Communications*, vol. 142, pp. 7–14, Feburary 1995.

[17] G. C. Hudek and D. J. Muder, "Signaling analysis for a multi-switch all-optical network," in *Proceeding of IEEE ICC*, 1995, vol. 2, pp. 1206–1210.

[18] J. Y. Wei and R. I. McFarland, "Just-In-Time signaling for WDM optical burst switching networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 2019–2037, December 2000.

[19] M. Yoo, C. Qiao, and S. Dixit, "QoS Performance of Optical Burst Switching in IP-Over-WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2062–2071, October 2000.

[20] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient channel scheduling algorithms in optical burst switched networks," in *Proceedings of INFOCOMM*, 2003, vol. 3, pp. 2268–2278.

[21] V. M. Vokkarane and J. P. Jue, "Segmentation-based non-preemptive scheduling algorithms for optical burst-switched networks," in *Proceedings of First International Workshop on Optical Burst Switching (WOBS)*, October 2003.

[22] V. M. Vokkarane and J. P. Jue, "Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1198– 1209, 2003.

[23] A. Detti, V. Eramo, and M. Listanti, "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)," *IEEE/OSA Journal of Lightwave Technology*, vol. 20, no. 2, pp. 154–165, 2002.

[24] X. Wang, H. Morikawa, and T. Aoyama, "Priority-based wavelength assignment algorithm for burst switched photonic networks," in *Proceeding of Optical Fiber Communication Conference*, 2002, pp. 765–767.

[25] Y. Chen, M. Hamdi, and D. H. K. Tsang, "Proportional QoS over OBS networks," in *Proceeding of GLOBECOM*, 2001, vol. 3, pp. 1510–1514.

[26] K. Dolzer, "Assured Horizon - A new Combined Framework for Burst Assembly and Reservation in Optical Burst Switched Networks," in *Proceeding of 7th European Conference on Networks & Optical Communications*, 2002.

[27] M. Jeong, H. Cankaya, and C. Qiao, "On a new multicasting approach in optical burst switched networks," *IEEE Communications Magazine*, pp. 96–103, November 2002.

[28] A. Detti and M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks," in *Proceedings of INFOCOMM*, 2002, vol. 3, pp. 1803–1812.

[29] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104–114, September 2000.