A Zero Burst Loss Architecture for star OBS Networks

Xenia Mountrouidou*, Vishwas Puttasubbappa**, Harry Perros*

*Computer Science Department,
North Carolina State University,
Raleigh, NC 27695, USA
{pmountr, hp}@csc.ncsu.edu
**Ericsson IP Infrastructure,
920 Main Campus Drive, Suite 500
Raleigh, NC 27606, USA
vishwas.puttasubbappa@ericsson.com

Abstract. Performance studies point to the fact that in an OBS network, the link utilization has to be kept very low in order for the burst loss probability to be within an acceptable level. Various congestion control schemes have been proposed, such as the use of converters, fiber delay lines, and deflection routing. However, these schemes do not alleviate this problem. It is our position that in order for OBS to become commercially viable, new schemes have to be devised that will either guarantee zero burst loss, or very low burst loss at high utilization. In a previous paper [1], we described effective zero burst loss schemes for OBS rings. In this paper, we present a zero burst loss scheme for star OBS topologies. Further research into the topic is required.

1 Introduction

Optical Burst Switching provides a good solution for transporting bursty traffic in an all optical network. The fundamental unit in an OBS network is a burst: a collection of packets grouped into a size that may vary according to the characteristics of the specific network. The most attractive feature of OBS is that it is all-optical; meaning, there is no OEO conversion of data within the OBS network. This characteristic reduces the overall system cost, but more importantly, offers a high speed and transparent network, independent of technology or data rate.

An OBS network consists of end-devices that we refer to as edge nodes. Edge nodes can operate both as transmitters and receivers of bursts. These devices are connected to various electronic packet-switched networks, such as IP, ATM and frame relay, and they also have one or more OBS interfaces. Each edge node is connected to one or more core OBS node which are interconnected through a mesh network. Each core node is an all bufferless optical cross connect (OXC). This means that the burst data are transmitted optically all the way

to their destination. Multiple bursts can be transmitted onto the same fiber simultaneously, since each fiber carries W wavelengths.

The main characteristic of an OBS network is the separation between data and control planes. Payload data is received and assembled into data bursts at each source edge node in the electronic domain, transported through one or more optical core nodes in the optical domain, and delivered to sink edge nodes where they are converted back to the electronic domain and disassembled into their constituent data packets for delivery to respective data sinks. In order to transmit a burst, a connection has to be established through the bufferless optical network. This is done by sending a control packet (also referred to as the setup packet in this paper) that includes information such as: source address, destination address, and duration of the burst. The control packet is transmitted optically in-band or out-of-band or it is transmitted electronically out-of-band, and it is processed by each core node electronically.

Another feature that distinguishes an OBS network from any other optical network is that the transmission of data is performed in bursts. The burst aggregation algorithm that is used to formulate the burst shapes the traffic in the OBS network. There are several algorithms for burst aggregation in the current literature. These algorithms consider a combination of the following parameters: a pre-set timer, a maximum burst size and a minimum burst size. When the timer expires, an edge may form a burst. Burst aggregation algorithms may offer QoS by adjusting their characteristics, such as timeout and/or minimum/maximum burst sizes corresponding to the traffic demand [2], [3].

Various resource reservation schemes have been proposed for the transmission of a burst (see Perros [4]). One of these schemes is *on-the-fly connection setup* with *delayed setup* and *timed release*. In the on-the-fly connection setup a control packet is first sent and then after a predetermined offset the corresponding data burst is sent. An OXC allocates the necessary resources within its switch fabric so that as to switch the incoming burst at the time the burst is due to arrive (delayed setup) for a period of time equal to the burst duration (timed release).

The on-the-fly connection setup scheme, which is the prevalent scheme, leads to burst loss. This is because, the setup request may be refused by an OXC due to contention at the required output port. However, this may not be known to the edge node at the moment of transmission of the burst. Burst loss is a negative characteristic in a high speed OBS network that promises to deliver QoS.

Several solutions to alleviate the problem of burst loss have been proposed such as fiber delay lines (FDL), wavelength conversion and deflection routing. A small number of FDLs ([5], [6], [7]) could be used in order to reduce burst loss. Fiber delay lines require lengthy pieces of fiber, and therefore they cannot be commercialized. Wavelength conversion is a viable solution to the burst loss problem. In this case, an incoming burst on a wavelength that is currently in use at the destination output fiber, can be converted to another free wavelength. Finally deflection routing ([8], [9]) may offer an alternative path to the destination device and divert a burst that would be lost otherwise. This path may include

more hops making deflection routing an ineffective method. Also, bandwidth has to be reserved especially for the overflow traffic over the path that the deflected burst will take.

Obviously in order for OBS to become commercially viable, new schemes have to be devised which will either guarantee a zero burst loss or a very low burst loss at high utilizations. In [10] we described zero burst loss access protocols for OBS rings that are efficient and they can also provide QoS for different classes of customers, such as HDTV streaming, non-real time high priority variable bit data, and best effort data. In this paper, we describe a zero burst loss scheme for star OBS networks. Obviously, more research along these lines is required.

The paper is organized as follows. In Section 2, we review various congestion control schemes that have been proposed for OBS networks. These schemes do not alleviate the problem of burst loss. Our zero burst loss scheme is described in Section 3. Results related to the performance of this scheme are given in Section 4. Finally the conclusions are given in Section 5.

2 Congestion Control Schemes

Congestion control in Optical Burst Switching (OBS) networks is an important research area. It is well known that OBS networks are prone to high burst losses and congestion can push these burst losses to alarming proportions. Although congestion control and congestion avoidance is an over-studied topic when it comes to IP and ATM networks, it poses several new and unresolved questions for OBS networks. Since an OBS network functions largely independent of any electrical or optical buffers at its core, congestion control schemes that are applied to buffered networks like ATM differ considerably in their architecture to that of OBS networks.

It is a well recognized fact that acceptable blocking rates in OBS mesh networks can be achieved only when the links are under-utilized. For example, in Figure 1 we give a plot of the blocking probability against utilization. This plot was obtained using both analytical and simulation models (see [10] for details) under the following assumptions. The graph is for a single outgoing wavelength in a core OBS node with 20 input and output fibers, 100 wavelengths per single fiber, 20 converters (i.e. 20% conversion rate), and with the ability for partial wavelength conversion with degree of conversion 20. Such an under utilized wavelength does not carry a high appealing factor for service providers who would always want to run their links heavily loaded.

In an OBS network, congestion and contention are closely related aspects. The boundaries that distinguish these two ideas are quite ambiguous. It is our conviction that contention and congestion drive each other, but the former is more transient in nature than the latter. Contention leads to burst loses due to lack of wavelengths at a core OXC. But the bursts arriving immediately after the losses might just pass through fine. Congestion occurs over a longer time frame and leads to increased contention problems over a longer time period.

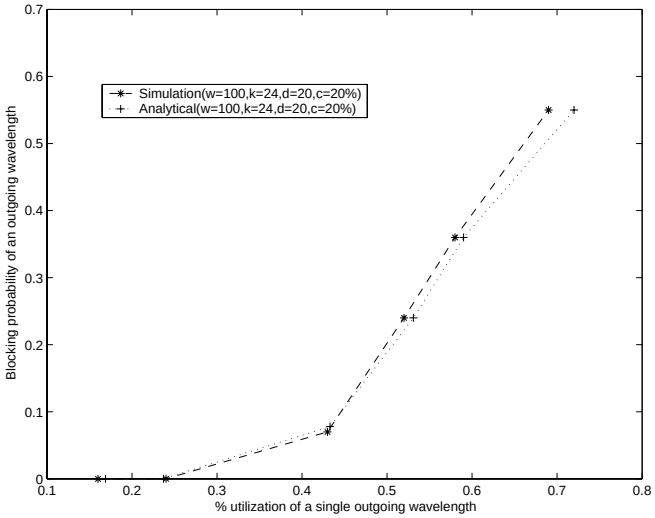

Fig. 1. Blocking vs Utilization

Research in OBS hitherto has mainly focused on contention resolution and there have been few contributions that have focused on congestion control. Several different approaches have been investigated in the literature including TCP -based approaches [11] [12], load balancing [13], alternate routing and deflection routing [14], [15], [8]. Below, we review some congestion avoidance schemes. For a more detailed discussion, please refer to Puttasubbappa [16].

2.1 Deflection routing

Deflection routing at a core OXC involves the selection of a different output fiber than the intended one, if the burst cannot be switched through the original output fiber. Deflection routing has its advantages and disadvantages.

1. The offset needs to be recalculated since the deflected burst takes a different path.
2. Offset recalculation will require intermediate nodes to be equipped with FDLs to delay the deflected bursts if necessary.
3. Deflected bursts may arrive out of order at the destination. End nodes may thus have to store large amounts of data.
4. Deflected bursts may end up in a loop and never reach their destination
5. There have been studies [17] [18] which indicate that deflection routing is ineffective at high traffic loads.

Deflection routing in OBS networks requires that the core nodes are equipped with FDLs. Using FDLs, a burst can be delayed for an extra amount of offset time and then sent over the deflected path. It has to be noted that the process

of deflection and offset recalculation can happen several times along the journey of a burst. A core node on receiving a setup message looks at its routing table to determine the next hop based on the destination information the setup message carries. Since the node has knowledge about the current usage of the wavelengths on each outgoing link, it can determine whether this particular outgoing link is congested. The core OXC may maintain not just the primary next hop routing entry but also secondary and tertiary next hops. Thus, each node based on its local information can choose any of the other next hops in case the primary is congested. It then has to calculate the additional offset value in case the route now chosen is longer (i.e. more hops) than the one the burst was traversing in. (This of course assumes that it knows how many hops the new route consists of). This additional delay is made up using FDLs. In case there is an absence of sufficient FDLs available to delay the burst, the burst will have to be dropped.

Some of the issues arising out of this mechanism are:

1. The amount of FDLs may be limited.
2. There are publications [17] [18] that deal with several aspects of this method. Key points arising out of these studies are:
   (a) Excessive deflection may lead to a congestion collapse
   (b) Excessive deflection may lead to longer end-to-end delays
   (c) Deflection routing may lead to re-ordering of bursts since bursts may take different routes, and thus higher layer protocols like TCP might find it difficult to operate optimally.

Absence of any optical buffering in the network complicates things in the sense that a burst once released with an offset value cannot be slowed down. Intermediate core nodes have no way of manipulating the offset and thus little can be done to prevent a burst loss in the presence of congestion. An alternative solution to the use of FDLs is to set all offsets to a value that is an upper bound of all offsets and it is such that the burst can be deflected any number of times (assuming no cycles), but still stays behind the setup message. This method may lead to possible under-utilization of the network.

Deflection routing itself can be implemented using several strategies found in protection and restoration of networks, such as:one hop deflection, path deflection, and N:1 deflection. In one hop deflection, each core node maintains next-hop primary, secondary and tertiary routes for a packet heading towards a particular destination. In the presence of congestion on an outgoing link, an alternative next hop is chosen by the core node. In path deflection, each core node calculates primary, secondary and tertiary paths to each destination. When the outgoing link of the primary (secondary) path gets congested, the core node chooses the outgoing link of the secondary (tertiary) path and this path has to be followed to the destination. Path deflection can either be implemented through source routing or using pre-establish GMPLS paths (LSPs). In bypass $N:1$ deflection, a congested link of a group of N links can be bypassed by using an alternate link, similarly to the $N:1$ technique in protection and restoration of networks.

It was shown by simulation that deflection routing is not an effective means to lowering the burst loss which can be greatly reduced with a few converters with restricted converter capabilities. (see Jonandula [19])

2.2 Feedback based schemes

This is a subject that has been studied quite extensively, e.g. the ABR scheme in ATM networks. Feedback messages relay the bandwidth usage and utilization of links back to the ingress OBS nodes thus enabling them to harness the dynamic state of the network. The feedback messages can either be sent as separate control messages or can be piggybacked to control messages traversing in the opposite direction resulting in minimization of control messages.

Feedback messages can be used to assist deflection routing. Specifically if OXCs know the utilization levels of links ahead of them, they can deflect bursts away from a congested link. Feedback mechanisms can also be used to determine the rate of burst transmission by the sources based on the congestion levels in the links the bursts are supposed to traverse. A feedback based setup has been studied for congestion-based routing techniques in [14]. Such a feedback based based routing technique has been shown to reduce loss probabilities.

2.3 Path recalculation

The congestion control techniques described in the previous sections can be seen as short-term schemes since they operate at smaller time scales. Path recalculation can be seen as a long-term congestion control scheme since it operates at a much larger time scale than the above schemes.

The motivation for path recalculation is that the state of the network in terms of congestion might reach a stage when short-term schemes can no longer be effective. In such a scenario, a radical change in routing paths needs to be made at a larger topological area.

The path calculation can either be distributed or centralized with a master node. Irrespective of the routing architecture, this mechanism facilitates a new routing pattern and thus a chance for the stagnant network to solve its congestion problems. Different source-destination flows between all pairs of nodes that satisfy quality of service criterion of the optical signal can be calculated.

3 A zero burst loss scheme for star networks

As shown in Figure 1, presented in the previous section, the utilization per wavelength has to be kept extremely low in order for the burst loss to be within an acceptable level. Congestion control schemes, such as deflection with FDLs, do not lower significantly the burst loss (see Jonandula [19]). As mentioned above, in order for OBS to become commercially viable, we will need schemes which either eliminate burst loss all together, or provide a very low burst loss but high utilizations.

In this section, we discuss a zero burst loss solution for star OBS networks. Current technological developments permit the transmission of an optical signal over a long fiber without intermediate amplification or signal restoration. This trend, obviously, is only going to continue in the future. In view of this, it is not hard to imagine that a single OBS core node can serve edge nodes over a large geographical area, whereby an edge node may be as many as 1,000 kilometers away from the core node. If the density of edge nodes is high, then multiple OBS core nodes can be used, as shown in Figure 2. In this case, each edge node has one OBS interface for each core node it is connected to. In the remaining of this paper, we will assume a single OBS core node, since the additional OBS nodes are independent of each other.

In a star configuration, it is possible to provide zero burst loss, if we let the core node do the scheduling. That is, the offsets are determined by the core node. Each edge node sends a control packet to the core node requesting a transmission of a certain duration to a particular destination edge node directly connected to the core node. Using a simple horizon scheduler, for each outgoing fiber the core node can manage all the burst transmissions without any loss. However, the propagation delays for far away edge nodes may take this toll on the network throughput. Below we describe the bimodal burst switching architecture that provides a solution to the issue of long propagations.

3.1 Bimodal Burst Switching Architecture

This architecture referred to as Bimodal Burst Switching (BBS) uses the delayed setup timed-release scheme compound with two modes of operation. The first mode (Mode 0) applies to an edge node that is close to core node, and the second mode (Mode 1) applies to a distant edge node. BBS can cover a large geographical area (1,000 km radius) and it is implementable in hardware as described in [20]. Also, it assumes that the OBS core node is a bufferless switch equipped with full conversion on each outgoing fiber.

Each edge node is linked to the core node via an upstream and a downstream fiber, each carrying W wavelengths. Furthermore, each edge node may use both operation modes to send data, depending on its proximity to the core node. Therefore, in a network that has N edge nodes, each edge node may include $2N-2$ destination queues, where packets are classified based on the destination and the mode of operation. The core node has a number of parallel switching planes equal to W, the number of wavelengths in a WDM link. The number of edge nodes that the core node can support is equal to the number of dual ports per switch plane.

This architecture introduces an innovative scheduler that performs the flow regulation of the traffic that arrives at each edge node. The scheduler is embedded in the controller of the core node. The core node does not use buffers on either inputs or outputs. The main structures that are used by the

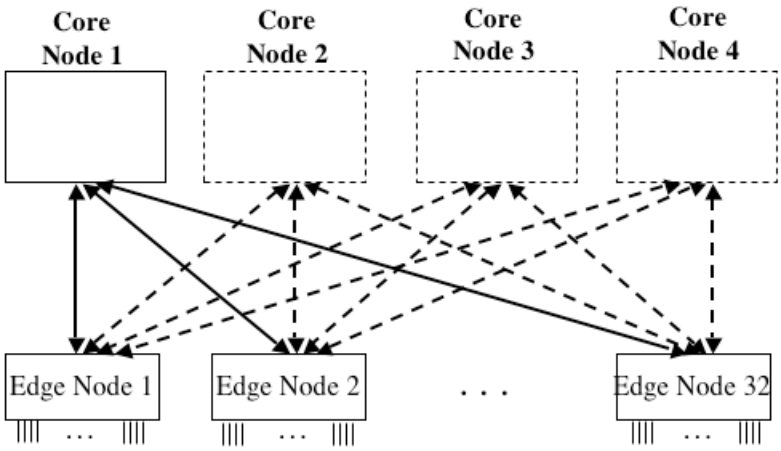

Fig. 2. Star OBS Network topology

controller in order to make a scheduling decision is the *Calendar* and the *M-element array*. The Calendar keeps track of the time when the uplink wavelengths to each edge node are going to be free. It consists of K elements. Preferably K is equal to $N * W$, where W is the number of wavelengths per fiber and N the number of edge nodes. We may also use a two-dimensional array with N rows and W columns to store the Calendar structure. If an element that belongs to the i^{th} row and w^{th} column of the Calendar is equal to j, then this means that the w^{th} wavelength of the edge node i is free at time slot j. The time slot of the Calendar structure is long enough to allow a contiguous period of time to schedule bursts and short enough to have small delays between bursts. It is usually between $1msec$ and $1\mu sec$. The M-element array keeps track of the availability of the output edge nodes. It consists of M elements, where M is $N * W$. We use a two-dimensional array with N rows and W columns for the M-element array elements, that are stored in the same way as the Calendar elements.

The scheduling algorithm, implemented by the controller, consists of two modes, Mode 0 and Mode 1. This differentiation is based on the proximity of the edge node to the core node. The proximity is determined by measuring the round-trip propagation delay between the edge node and the core node. Mode 0 is used for edge nodes that are at a small distance d from the core node ($d \leq 100\ km$) and consequently have a small round trip propagation delay. On the other hand, Mode 1 is more suitable for distant edge nodes ($d > 100\ km$) that have a large propagation delay. As will be explained below the main difference between these two modes is that in the first mode the flow rate regulation is provided to waiting bursts whereas in the second mode it is provided to anticipated bursts.

Mode 0 scheduling. In this case, an edge node sends requests to transmit bursts to its ingress OXC. These requests are sent by the edge node at fixed intervals. The operation of Mode 0 is as follows:

- **Transmission:** Edge node i receives packets which it then buffers to the appropriate destination queues. In an OBS network with N edge nodes, there exist $N-1$ Mode 0 destination queues at each edge node. Every T μsec, the edge node checks the input queues and forms bursts subject to a minimum and a maximum burst size. For each burst it issues a burst request that is stored at the burst request queue. Each edge node has one request queue where it stores all the burst requests for any destination, until they are sent to the core node. Each request consists of a number of fields such as: source, destination, size and an ID number.
- **Scheduling:** Every T μsec edge node i sends all the burst requests it has stored until that moment in a single control packet to the core node. This procedure needs time equal to one-way propagation delay to be completed. Once the control packets reach the core node, the controller that implements the scheduler decides when the burst will be transmitted using the *shortest horizon* scheduling policy. This decision is formed using the Calendar and the M-element array. The scheduler scans the Calendar to find the first uplink wavelength of any edge node that is free, then calculates the horizon for the specific edge node's requests. The horizon for each burst request is computed as the difference of the time slot at which any downlink wavelength of destination is free, to the time slot that the uplink wavelength of the source is free. (Full wavelength conversion is assumed). The burst that is destined to the edge node which has the minimum horizon value is served first. According to the proposed scheduling policy it is preferred to schedule the minimum negative value, because this means that the destination is available earlier than the source, so the source may start transmitting immediately. For example, if source edge node i has a free uplink wavelength at time slot 10 and requests to transmit to destination edge nodes j and k that have free downlink wavelengths at times 5 and 15, the horizons are -5 and 5. It is preferable to schedule the request destined to j since the source can start transmitting to it immediately.

  After a request is served the Calendar and the M-element array are updated. Then the Calendar is scanned in order to find the next available wavelength and the above scheduling procedure is repeated, until all the requests that the edge nodes sent at this time are scheduled. When all the requests are scheduled, the core node sends permits to the edge nodes containing information as to when they can transmit their bursts. An edge node may receive permits on its downlink wavelengths while at the same time it is sending new requests on its uplink wavelengths. Interleaving burst requests and permits reduces the waiting time of a request.

  It is clear that this mode depends highly on the round trip time. The delay of a burst request is equal to one round trip time plus the queueing delay.

The queueing delays of a particular burst request depend on the number of requests that are scheduled prior to this.
- **Reception:** Each destination edge node receives bursts from the core node which are buffered electronically, disassembled to packets and then delivered to its users through other interfaces.

Mode 1 scheduling. In Mode 1 burst switching, data is still transmitted in bursts, but the initial phase in OBS where an edge node sends a request to its ingress OXC has been eliminated. Mode 1 scheduling is preferable when the propagation delay is large. Unlike Mode 0, a Mode 1 edge node does not issue burst requests. Rather, the edge node requests and is allocated a fixed bandwidth for each destination during the initialization phase. This bandwidth is calculated based on the traffic between the edge node and each destination edge node, and it is made available to the edge node in the form of burst sizes. These burst sizes are fixed in time and they repeat periodically.

Let t_{ij} μsec be the transmission time allocated to the traffic from i to j, and let this be repeated every T μsec. Then the bandwidth allocated to edge node i for transmitting traffic to edge node j is $(t_{ij}/T)*V$ $Gbits/sec$, where V $Gbits/sec$ is the transmission speed. The edge node communicates the values t_{ij}, $j = 1, ..., N$, $j \neq i$ and T to the controller. The controller issues automatically a burst request of duration t_{ij} every T μsec for each destination, where $t_{ij} \leq T$ for every $i, j = 1, 2, ...N$. These burst requests are then scheduled following the same procedure as in Mode 0 operation. Next the scheduler issues permits which are sent to the edge node. It is clear that the core node defines a different bandwidth allocation for every stream ij. This offers a flexibility to satisfy the different traffic requirements of each stream. We note that the bandwidth allocated to each Mode 1 edge node by the controller can be renegotiated using specially designed messages. Such renegotiation will take place when the traffic arriving at an edge node changes significantly. Adapting the bandwidth allocation to the traffic demand is considered as a congestion avoidance scheme. Therefore, the BBS architecture prevents burst loss due to congestion.

The Mode 1 operation is summarized as follows:

- **Transmission:** The edge node may transmit the data it has gathered up to this moment based on the permit information. In this case there is no minimum or maximum burst size used to define the size of a burst. This means that the burst aggregation algorithm used at the edge nodes does not have an effect in the burst characteristics when using Mode 1. The burst size B is defined by the transmission time t_{ij} as: $B <= t_{ij} * V$ $Bytes$. When a Mode 1 edge node i receives a permit it will transmit data for the duration t_{ij}. Assume, for instance, that the data it has requires 112 μsec to be transmitted. Assume also that $t_{ij} = 100$ μsec. Then in this case, it will not be able to transmit all the data, and 12 μsec worth of data will remain in its buffer. On the other hand, if it has 80 μsec worth of data, then it will transmit all its data and the remaining 20 μsec of the t_{ij} period will be unused.

- **Scheduling:** The controller creates a burst request of duration t_{ij} for every destination $j \neq i$, and for every Mode 1 edge node i, every T units of time. These requests are then placed at the scheduler's queue, and they are scheduled in the same manner as Mode 0 burst requests. Notice that the burst requests are generated by the controller and not by the edge nodes. Transmission permits are then sent to the Mode 1 edge nodes.
- **Reception:** The destination edge node j receives bursts which are buffered electronically, disassembled to packets and then delivered to its users.

The main difference between the two scheduling modes is that in Mode 0 already existing bursts are scheduled whereas in Mode 1 permits for anticipated bursts are issued. Generating fixed size requests for every edge node requires no knowledge whether they have bursts to transmit. Also it does not require knowledge of the size of their packet queues. This may lead to bandwidth loss if the edge nodes do not have data to transmit to every destination, or if they have smaller queues than the fixed size that is set. Also it may lead to larger delays if they have larger bursts than the bandwidth allocated. This is why we may need to adjust the bandwidth allocation when the arrival traffic pattern at the edge nodes changes.

The operation of the bimodal scheduler is depicted in Figure 3. In the case of a nearby edge node (Mode 0), the edge node sends all the burst requests it has accumulated up to this moment every fixed period of time, say every 256 μsec. The core receives the requests, schedules them according to the shortest horizon scheme and then sends permits to the edge nodes. Finally, the edge nodes transmit their bursts according to the permits they received from the core. The fixed period used to send requests is short, and as a result it provides a continuous supply of permits to the edge nodes.

In the case of a distant node (Mode 1) the core node creates burst requests periodically which are then scheduled according to shortest horizon. When a Mode 1 edge node receives a permit, it transmits data for a fixed period of time. The main difference in this scheme is that there are no requests from the edge nodes to the core node. This provides a more efficient scheme since the one-way propagation is large.

4 Simulation results

In this section we evaluate the performance of the BBS architecture using simulation. N edge nodes and one core node were simulated. We assume that edge nodes 1 to $N/2$ are within a small distance d from the core node, where 10 km $<$ d $<$ 100 km, which means that they are served using Mode 0 scheduling. The remaining edge nodes $N/2+1$ to N are more than 100 km away, which means that the core node serves them using the Mode 1 scheduling mechanism. Burst aggregation is performed using timeout and minimum/maximum burst sizes. The minimum burst size and maximum burst size were fixed to 16 kB and 112 kB respectively.

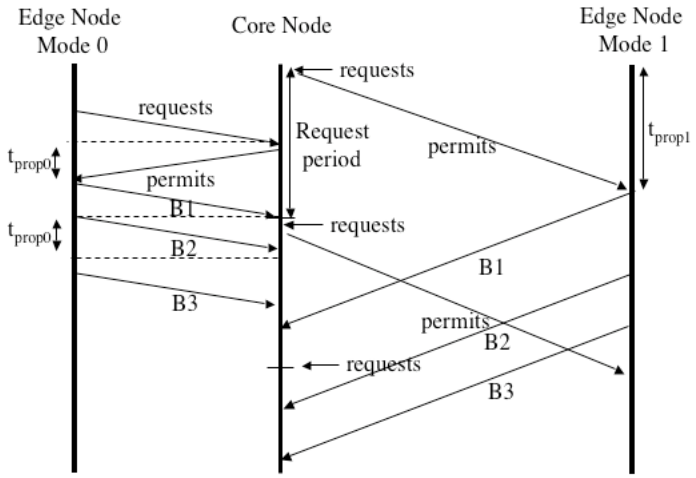

Fig. 3. The operation of the bimodal scheduler

Furthermore, the burst aggregation period T was set to 256 μsec. The same period T is used for convenience as a request period for Mode 0 edge nodes and as a permit period for Mode 1. The one way propagation delay between an edge node and the core node for Mode 0 edge nodes was set to 500 μsec, which means they are at a 100 km distance from the core node, and for Mode 1 edge nodes to 5,000 μsec, which means they are at 1,000 km from the core node. In this study we have assumed out-of-band signaling. The signaling messages can also be implemented in-band, but this was not considered here. Finally, renegotiation of the bandwidth allocation in Mode 1 scheduling is expected to take place less frequently compared to the time scales of the burst transmission operation, and it was not considered in our simulation study.

Each edge node has a 10 MB electronic buffer to store the packets that arrive from external sources. The arrival process is an Interrupted Poisson Process (IPP) as described in [21]. This IPP arrival process is an ON/OFF process, where both the ON and OFF periods are exponentially distributed. Packets arrive back-to-back using Poisson distribution with rate λ during the ON period. The transmission speed is 10 Gbps. Packets do not arrive during the OFF period. The packet length is assumed to be exponentially distributed with an average of 500 bytes. The last packet of the ON period may be truncated so that its last bit arrives at the end of the ON period. The squared coefficient of variation c^2 of the packet interarrival time was used to characterize the burstiness of the packet arrival process. This coefficient is defined as the variance of the packet inter-arrival time divided by the squared mean packet inter-arrival time. Assuming that the distribution of the ON period is exponential with average $1/\mu_1$ and the distribution of the OFF period is exponential with average $1/\mu_2$ we have:

$$c^2_{IPP} = 1 + \frac{2\lambda\mu_1}{(\mu_1 + \mu_2)^2}$$

where λ is the arrival rate of a packet during the ON period and $\frac{1}{\lambda} = \frac{(500Bytes)}{(10Gbps)} = 0.4\mu$sec. Finally to characterize completely the arrival process the *average arrival rate* is used, given by:

$$average\ arrival\ rate = \frac{(10Gbps)\mu_2}{\mu_1 + \mu_2}$$

Given the c^2 and the average arrival rate we calculate the quantities μ_1 and μ_2. In our simulation experiments c^2 was set to 5 and 20, and the arrival rate was varied from 6 Gbps to 100 Gbps. Packets arriving at an edge node were assigned to a destination using the uniform distribution. This arrival process captures the burstiness of the Internet traffic, especially when voice and video are transferred. It is also confirmed experimentally that it models accurately the traffic in a network [22], [23].

The simulation outputs consist of the mean overall delay per packet for all nodes and the percentage of utilization of an uplink or a downlink wavelength. In all the figures provided, the results are plotted with 95% confidence intervals estimated by the method of the batch means [24]. The number of batches is set to 30 and each batch consists of at least 10,000 bursts/edge node. The confidence intervals are very narrow and as a result are barely visible in the figures.

The Bimodal Burst Switching (BBS) scheme is compared against the case where all N edge nodes operate under the Mode 0 scheme, indicated in the graphs as "Mode 0". BBS is also compared against the case where all N edge nodes operate under the Mode 1 scheme, that is the bandwidth allocation scheme, indicated in the graphs as "Mode 1". We recall that in the BBS scheme edge node 1 to $N/2$ operate under Mode 0 and edge nodes $(N/2 + 1)$ to N under Mode 1. The calculation of the intervals t_{ij} for Mode 1 was based on the average arrival rate. Full wavelength conversion was assumed.

An overall picture of the delay per packet when all edge nodes are scheduled using the three scheduling schemes under study for c^2=5, is shown in Figure 4 (a). The average arrival rate at every edge node is 6 Gbps. The delay of a packet is the time elapsed from the moment it fully arrives at an edge node to the moment it is delivered to a destination edge node. That is, it consists of the queueing delay at edge node plus the propagation delay from the transmitting edge node to the destination edge node. Edges nodes 1 to 5 are at short distance from the core node (i.e. they have 500 μsec one-way propagation delay) and edge nodes 6 to 10 are far away (i.e. they have a 5,000 μsec one-way propagation delay). Packets arriving at each edge node were assigned to a destination node using the uniform distribution. The increased delays of the traditional OBS scheme can be easily seen in this figure. As the number of wavelengths increases, we observe an almost linear decrease in the packet delay for the BBS and Mode 1 schemes. The difference between Mode 0 and BBS evident: Mode 0 overall average delay per packet is much higher. An interesting observation is that the average delay per

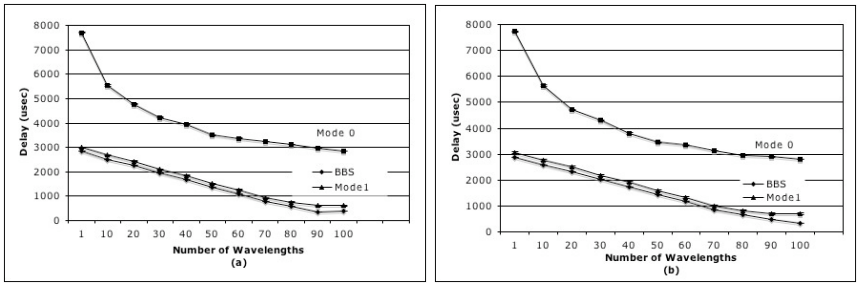

Fig. 4. (a) Mean packet delay for all 10 edge nodes vs. number of wavelengths for c^2=5, (b) Mean packet delay for all 10 edge nodes vs. number of wavelengths for c^2=20

packet for the BBS and the Mode 1 are very close. This leads us to the conclusion that differentiating our scheduling technique between distant and closeby edge nodes does not offer a large improvement on the average delay per packet. Figure 4(b) gives the average delay per packet when the input traffic is burstier, i.e. $c^2 = 20$. This burstiness corresponds to traffic that may have long intervals of silence (OFF period), like VoIP or video. It can be observed that there is no significant difference for the BBS and Mode 0 schemes when the burstiness increases.

Figure 5 shows the percentage of utilization of an uplink/downlink wavelength. The uplink and downlink wavelength utilization is the same. That is because the same arrival process to each edge device was assumed and destination nodes are uniformly chosen. When we use only one wavelength in our model, wavelength utilization approaches 60%. All three schemes have the same utilization. As mentioned above Mode 1 scheduling scheme is used to schedule bursts that are not yet formed at the edge node. If the wavelength utilization is high, this means that there is always a burst formed for each destination in every edge node that is scheduled using this scheme. Then the bandwidth that is allocated periodically is not wasted. On the other hand, the utilization per wavelength decreases since the number of wavelengths increases and there are more alternative paths for a data burst. This means that lower per wavelength utilization does not affect Mode 1 and BBS schemes if the input traffic and the overall utilization remains the same (60% for all wavelengths in one fiber link).

Figure 6 shows how the average delay per packet is affected when the average arrival rate of the input traffic is varied and the rate of packet arrivals is set to 100 Gbps for all three scheduling schemes, with all other parameters remaining the same as above. There is one uplink and one downlink wavelength for every fiber link. When the average arrival rate is >80 Gbps we get very high delays for Mode 0 and BBS, whereas Mode 1 gives very high delay when it is >90 Gbps. These delays are not drawn in this Figure. The BBS and Mode 1 schemes scale well when the average arrival rate increases. Mode 0 on the other hand has a high increase in the mean delay when the average arrival rate is >60 Gbps. This proves that BBS and Mode 1 are suitable for the high bandwidth demands.

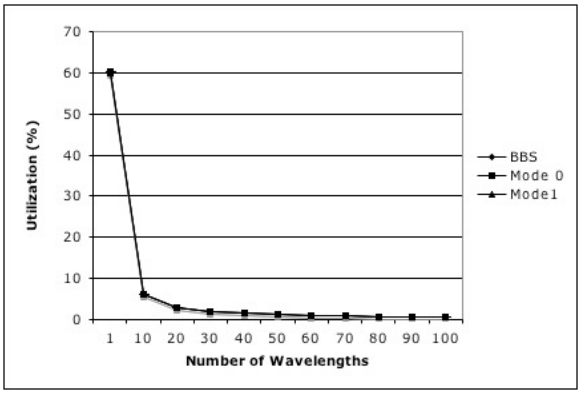

Fig. 5. Mean utilization for all 10 edge nodes vs. number of wavelengths

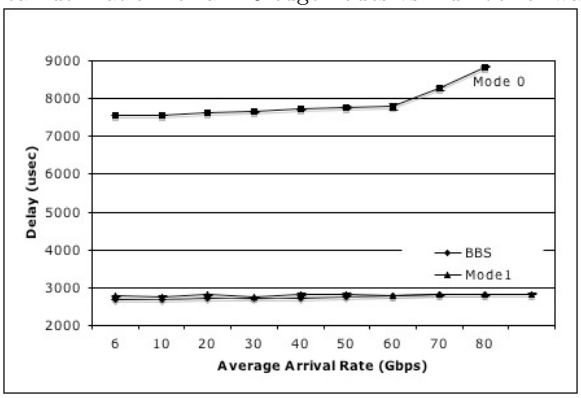

Fig. 6. Mean packet delay for all 10 edge nodes vs. average arrival rate

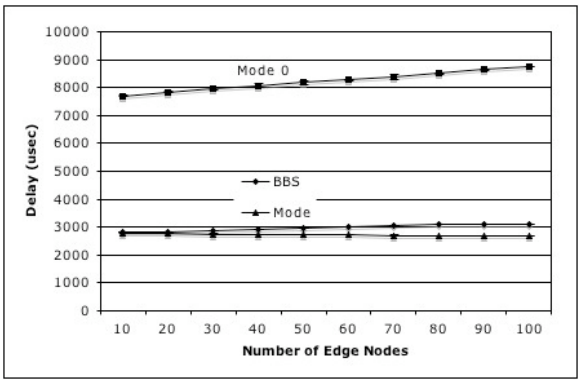

Fig. 7. Mean packet delay for all edge nodes vs. number of edge nodes for c^2=5

In Figure 7 the average delay per packet for all edge nodes is plotted when the number of edge nodes for $c^2 = 5$ is varied. It is assumed that W = 1, i.e. one uplink and one downlink wavelength. The BBS scheme scales well as the number of edge nodes increases, whereas Mode 0 has large delays. It is also observed that Mode 1 scales very well, remaining almost constant. The low delays of Mode 1 and BBS is contrasted to the high utilization percentage, that is about 60% when only one wavelength is used.

At this point the limitation of Mode 1 compared to BBS is exposed. Based on our simulation experiments both schemes have similar performance, therefore one would think that there is no point in differentiating scheduling in two modes. Mode 1 scheduling would be efficient to schedule all nodes. This is not the case. Mode 1 requires bandwidth allocation and when the edge nodes increase in number we have to increase the bandwidth allocated to each one of them as well. This may not be feasible on a link with finite bandwidth. On the other hand BBS is using Mode 0 in combination with Mode 1. Therefore, it does not need to allocate bandwidth for all edge nodes, but only for those that are scheduled using Mode 1. Furthermore Mode 1 scheduling scheme may waste useful bandwidth, as it is static most of the time. When the input traffic is low, the bandwidth allocated by Mode 1 scheduling may be too large, and therefore it will be wasted. On the other hand if the traffic is too high, there will be large delays as the packets will have a longer queueing time. Therefore, if the traffic pattern changes oftenly, Mode 1 scheduling is not efficient. Further work on the evaluation of the BBS scheme, can be found in [25].

5 Conclusions

In order for OBS to be commercially viable, schemes have to be devised that provide very low burst loss at high utilizations or are burst loss free. In this paper, we presented a burst loss free scheme for star OBS networks. More than one star OBS network can be used in order to provide large geographic coverage. How

these separate start networks can be linked together so that the entire resulting network is burst loss free, is a problem currently under investigation.

References

1. Puttasubbappa, V., Perros, H.: Access protocols to support different service classes in an optical burst switching ring. (In: Networking 2004)
2. Choi, J., Choi, J., Kang, M.: Dimensioning burst assembly process in optical burst switching networks. In: IEICE Transactions on Communications. Volume E88-B. (2005) 3855–3863
3. Vokkarane, V., Jue, J., Sitaraman, S.: Burst segmentation: an approach for reducing packet loss in optical burst switched networks. In: IEEE International Conference on Communications. Number 5, IEEE (2002) 2673–2677
4. Perros, H.G.: Connection-oriented networks: SONET/SDH, ATM, MPLS, Optical Networks. Wiley (2005)
5. Yoo, M., Qiao, C., Dixit, S.: Qos performance of optical burst switching in IPover-WDM networks selected areas in communications. IEEE Journal on Areas in Communications **18**(10) (October 2000) 2062–2071
6. Gauger, C., Dolzer, K., Scharf, M.: Reservation strategies for FDL buffers in OBS networks. In: Proceedings of the IEEE International Conference on Communications, IEEE (2002)
7. Yoo, M., Qiao, C., Dixit, S.: The effect of limited fiber delay lines on qos performance of optical burst switched WDM networks. In: Proceedings of the IEEE International Conference on Communications. Number 2, IEEE (2000) 974–979
8. Hsu, C.F., Liu, T.L., , Huang, N.F.: On the deflection routing in QoS supported optical burst-switched networks. In: IEEE International Conference on Communications. Number 5, IEEE (2002) 2786–2790
9. Kim, S., Kim, N., , Kang, M.: Contention resolution for optical burst switching networks using alternative routing. In: IEEE International Conference on Communications. Number 5, IEEE (2002) 2678–2681
10. Puttasubbappa, V., Perros, H.: An approximate queueing model for limited-range wavelength conversion in an OBS switch. (In: Networking 2005)
11. Wang, S.: Using TCP congestion control to improve the performances of optical burst switched networks. IEEE ICC'03 (International Conference on Communication) (2004)
12. Wang, S.: Decoupling Control from Data for TCP Congestion Control. PhD thesis, Harvard University (1999)
13. Battiti, R., Salvadori, E.: A load balancing scheme for congestion control in MPLS networks. Technical report, Universita di Trento (2002)
14. Vokkarane,, G.T.V., Jue, J.P.: Dynamic congestion-based load balanced routing in optical burst-switched networks. IEEE Globecom (2003)
15. Kim,, S.K.N., Kang, M.: Contention resolution for optical burst switching networks using alternative routing. In Proceedings of IEEE ICC (2002)
16. Puttasubbappa, V.: Optical Burst Switching: Problems, Solutions and Performance Evaluation. PhD thesis, North Carolina State University (2006)
17. Morikawa, X.W.H., Aoyama, T.: Deflection routing protocol for burst switching WDM mesh networks. In proceeding of Opticomm (2000) 257–266
18. Wong, A.Z.H.V.Z.R.E., Zukerman, M.: Reduced load Erlang fixed point analysis of optical burst switched networks with deflection routing and wavelength reservation. The First International Workshop on Optical Burst Switching (WOBS) (2003)

19. Jonandula, V.: Performance analysis of congestion control schemes in OBS mesh networks. MS thesis (2004)
20. Beshai, M.: Burst switching in a high capacity network. US Patent (2001)
21. Fischer, W., Meier-Hellstern, K.: The Markov-Modulated Poisson Process (MMPP) cookbook. Performance Evaluation **18** (1992) 149–171
22. Karagiannis, T., Molle, M., Faloutsos, M., A.Broido: A nonstationary poisson view of the internet. IEEE INFOCOM (2004)
23. Heffes, H., Lucantoni, D.M.: A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. Journal on Selected Areas in Communications **SAC-4**(6) (1986) 856–868
24. Perros, H.G.: Computer simulation techniques: the definitive introduction. Available at: http://www.csc.ncsu.edu/faculty/perros/books.html (2003)
25. Mountrouidou, X., Perros, H., Beshai, M.: Performance evaluation of optical burst switching schemes for grid networks. In: GridNets 2005, IEEE (2005)