Traffic Shaping and Scheduling for OBS-based IP/WDM Backbones

Mahmoud Elhaddad, Rami Melhem, Taieb Znati, Debashis Basak

Department of Computer Science, University of Pittsburgh, Pittsburgh, Pennsylvania 15260

ABSTRACT

We introduce Proactive Reservation-based Switching (PRS)—a switching architecture for IP/WDM networks based on Labeled Optical Burst Switching. PRS achieves packet delay and loss performance comparable to that of packet-switched networks, without requiring large buffering capacity, or burst scheduling across a large number of wavelengths at the core routers. PRS combines proactive channel reservation with periodic shaping of ingress–egress traffic aggregates to hide the offset latency and approximate the utilization/buffering characteristics of discrete-time queues with periodic arrival streams. A channel scheduling algorithm imposes constraints on burst departure times to ensure efficient utilization of wavelength channels and to maintain the distance between consecutive bursts through the network. Results obtained from simulation using TCP traffic over carefully designed topologies indicate that PRS consistently achieves channel utilization above 90% with modest buffering requirements.

Keywords: Optical burst switching, WDM networks, TCP, performance evaluation

1. INTRODUCTION

In recent years, research into building high-speed IP/WDM routers has exhibited a trend toward leveraging the advances in optical devices and fast optical switching to circumvent the capacity bottleneck and high power consumption in IP routers. One approach is optical packet switching (OPS), in which the packet payload is optically buffered using Fiber Delay Lines (FDLs) at each router while its headers are processed electronically to compute the forwarding decision. In addition to the limited buffering capacity achievable using FDL buffers, OPS poses switch scheduling and control problems that remain under research.[1]

A hybrid electro-optical approach is adopted in the Optical Router (OR) project,[2] which proposes substituting an optical switching fabric for electronic cross-bars in traditional routers. Still, packets are buffered and processed electronically at the line cards. To match the speed limitations of electronic line cards (≈ 40 Gbps), incoming WDM links are demultiplexed into single-wavelength fibers so that a line card handles traffic from only one wavelength channel, thus requiring a large number of ports per router.

A third approach, is Optical Burst Switching (OBS).[3–5] In an OBS network, IP packets are assembled into bursts at the ingress routers, and, ahead of each burst transmission, a burst header packet is forwarded on a dedicated control channel along the path of the burst. The header packets are processed electronically at every routers to compute the forwarding decision and to reserve router resources (e.g., output channel, FDL buffer, and wavelength converters) to the incoming burst. The resource allocations at a given router represent a schedule according to which arriving bursts are switched transparently through the router. The duration between the transmission of a header packet and its corresponding burst is called the *transmission offset* (simply, offset). The offset should be large enough so that the reservation request is processed at all routers ahead of burst arrival.

In this paper, we introduce Proactive Reservation-based Switching (PRS), an IP/WDM switching architecture that overcomes the limitations of OBS, particularly, the limited bandwidth utilization efficiency of data and control channels, and the increased packet delay compared to packet switching. The proposed architecture combines a proactive reservation scheme with periodic shaping of ingress–egress traffic aggregates. This combination hides the offset latency and approximates the utilization/buffering characteristics of discrete-time queues with periodic arrival streams. A channel scheduling algorithm

E-mail: {elhaddad, melhem, znati}@cs.pitt.edu, debashis@adelphia.net

imposes constraints on burst departure times to ensure efficient utilization of wavelength channels and to maintain the distance between consecutive bursts through the network. The goal of the proactive-reservation scheme in PRS is to allocate bandwidth to individual traffic aggregates according to a specified rate, much as TDM switching does. However, compared to TDM-based transparent switching, the PRS relaxes the synchronization requirements and allows the ingress routers to react to changes in bandwidth demands and network conditions by adapting the reservation rate.

The network utilization efficiency depends on the effectiveness of the burst scheduling algorithms used at the ingress and core routers. In *horizon scheduling*,[6] an incoming burst cannot be scheduled on a given channel before the latest-scheduled burst transmission on that channel, which leads to potentially large unused intervals (called *voids*) in the channel schedule, and hence poor channel utilization. Scheduling algorithms with void-filling have been proposed.[7–9] However, these schedulers may result in gaps between adjacently scheduled bursts that are too small to fit additional ones. The gaps are created because reservation requests have arbitrary start times. The ratio of the average gap size to the burst size represents a further reduction in effective channel capacity. PRS uses a slotted burst scheduling algorithm that eliminates the gaps by buffering bursts to align their reservation requests with channel slot boundary. The scheduling algorithm also maintains the periodicity of individual trunks through the network.

A performance limitation of OBS in comparison with packet switching is the increased packet latency due to burst assembly and offset delays incurred at the ingress routers. We exploit the predictability of periodically-shaped traffic to completely hide the offset latency through proactive channel reservation, in which the reservation requests for individual trunks are periodically generated by the corresponding ingress routers in anticipation of burst formation. In addition to improving packet delay, proactive reservation was motivated by the need to improve the efficiency of control channels. This is achieved in PRS by bundling multiple reservation requests over a short period of time into one reservation packet thus amortizing the headers overhead, which is significant especially in networks using an IP-based control plane.

The inefficiency of OPS and OBS in utilizing network capacity compared to electronic packet switching is due in part to an inherent trade-off between packet loss rate and network utilization that results from the limited FDL buffering capacity. In OBS, packet loss is a consequence of burst blocking due to contention at the output of core routers. The large variability in the arrival process of IP packet traffic at the ingress routers results in high burst blocking rate as bursty packet arrivals at the ingress nodes lead to periods of contention among trains of back-to-back bursts at the output OBS core routers followed by idle periods. The burstiness of IP traffic typically results in periods of back-to-back arrivals forming burst trains followed by prolonged idle periods. Coincidental synchronization among trains from different sources results in high loss rate. Spacing burst arrivals from the same source so that they do not compete among themselves for buffer capacity can significantly reduce the loss rate.

In order to reduce the loss rate, the chance of collision among trains of bursts needs to be reduced, hence the trade-off between loss rate and network utilization (or equivalently, offered load in $\mathrm{bursts/second}$). This effectively reduces the capacity of network channels. Although backbone links are usually lightly loaded under normal operation, the excess capacity is often planned to carry rerouted traffic in the case of link failure. In this paper, we show that regulating individual ingress–egress traffic aggregates (henceforth referred to as *trunks*) into periodic streams allows the efficient utilization of network capacity with modest buffering requirements.

The remainder of the paper is organized as follows. In the next section, we provide a brief overview of optical burst switching; in Section 3, we introduce the PRS architecture and describe the proactive reservation scheme. The case for periodic trunk shaping is presented and the PRS burst scheduler is introduced in Section 4. In Section 5 we report results from a simulation-based evaluation of the loss-utilization performance of PRS. Concluding remarks are given in Section 6.

2. OPTICAL BURST-SWITCHED NETWORKS

An OBS network consists of OBS nodes, also referred to as *optical core routers or simply as* core routers, and electronic edge routers connected by WDM links. Packets are assembled at network ingress into bursts. The bursts are then forwarded through the OBS network to the egress, where the bursts are disassembled back into packets. The edge routers must provide capability to assemble and disassemble bursts on interfaces facing the OBS network. On the other side the edge routers continue to support legacy interfaces such as SONET/SDH (POS) and Gigabit Ethernet. A core router consists of an optical switching matrix and a switch control unit.

Prior to sending out a burst into the OBS network, core routers must be configured to switch the burst. This is achieved by sending out a reservation request (control) packet slightly ahead of time. The control packet traverses the OBS network to reserves the channels along the path. The switch control unit on each core router has a controller that runs routing and signaling protocols. The reservation request may be routed through the OBS network in two ways. It may be routed hop by hop as in a connectionless IP network which requires an IP lookup for each control packet. Alternatively, label switched trunks may be established between edge routers, and the control packet then contains a label that determines how it is to be routed at a core router. The latter approach also referred to as Labeled Optical Burst Switching (LOBS).[10]

Establishing a traffic engineered trunk in LOBS is different from setting up an optical circuit in wavelength routed networks, for example using GMPLS. In LOBS, allocation of bandwidth to trunks can be at sub-wavelength granularity, thus allowing a wavelength channel on a given WDM link to be shared among multiple trunks. This is conceptually closer to establishing trunks in MPLS-based packet networks.

The initiation of the reservation process for a burst at the ingress or the arrival of the corresponding control packet at a core router triggers the scheduling algorithm for the outgoing link/wavelength(s) on which the burst needs to be forwarded. Since requests for transmission of a burst are generated asynchronously it is possible that two or more bursts intended for the same channel arrive in an overlapped fashion. This implies that a burst may need to be dropped or delayed to the time when it can be scheduled in a conflict-free manner. A burst may be delayed using FDLs.

To alleviate the loss-utilization trade-off, burst scheduling is typically performed across a large number of data channels (assuming wavelength conversion capability) on the desired output link to emulate a multi-server queue. As the output link scheduler needs to process a reservation request per burst, its service rate (in reservations/second) needs to be at least equal to the total link throughput in bursts/second. A slow scheduler would limit the link throughput. Given an electronic processing speed of few GHz, a simple calculation assuming a link of 32 wavelengths at 100Gb/s, an average burst size of 10 KB, and 10 clock-cycles per scheduling operation reveals that scheduling across the candidate wavelengths must be attempted in parallel. An associative-memory based hardware design for LAUC-VF – a void-filling scheduling algorithm that supports parallel scheduling was presented in Ref. 11.

3. PROACTIVE RESERVATION-BASED SWITCHING

PRS is based on a label-switched control architecture to facilitate traffic aggregation at the ingress nodes into traffic trunks based on the egress address and quality of service requirements. Traffic aggregation allows for efficient burst reservation and traffic shaping. The sequence of labels that a reservation packet assumes as it traverses the network links determine the label-switched path (LSP) for the corresponding trunk. For reasons of forwarding efficiency, LSPs may be merged as they reach the first of a sequence of common links, then later split as their paths diverge. A trunk identifier within the reservation packet payload enables the switch controllers to perform burst scheduling based on trunk membership.

A Routing and Wavelength Assignment (RWA) algorithm is used to compute for each trunk a path and a set of wavelengths that it can use on each link along that path based on its expected demand and the wavelength conversion constraints at the core routers. This algorithm is run whenever there is a change in the network topology or traffic demands.

PRS regulates trunks into periodic streams to achieve high network bandwidth utilization without large buffering requirements at the core. Since the ingress routers are electronic packet switches, large buffers can be used to smooth the bursty packet arrival process without incurring substantial packet loss. Periodic traffic regulation (periodic shaping) effectively implements trunk policing based on peak rate allocations. In case of an efficient network-wide bandwidth allocation, no flow is allocated bandwidth more than it demands. To achieve a high degree of statistical multiplexing under periodic shaping, the bandwidth allocations should be dynamically revised to maintain efficiency. In addition to efficiency, a common objective of bandwidth allocation is fairness. Consider two trunks competing for bandwidth at a bottleneck, the trunk with a higher number of flows should be allocated more bandwidth–assuming that individual flows within each trunk have demands higher than their corresponding allocations. Techniques for the estimation of fair and efficient bandwidth allocations at the level of traffic aggregates were proposed in Ref. 12 (TCP trunking) and in Ref. 13. These techniques attempt to estimate the instantaneous fair allocation for each trunk so that network capacity is efficiently utilized.

Dynamic bandwidth allocation may interfere with periodic reservation in case the bandwidth allocations changes over short time scales. To avoid this problem, the time averages of the instantaneous allocations can be used as the peak rate allocation for the trunks. It has been repeatedly observed[14–16] that the number of flows crossing individual links at the Internet's core exhibits little variability over relatively large intervals of time (minutes). This implies little variability in the time average of the allocations produced by the above techniques over intervals that are much larger than the trunk reservation cycle.

3.1. The case for proactive reservation

Due to the burst assembly and transmission offset latencies, The average packet delay along an OBS network path is typically higher than that along an identical path in a packet-switched network. Large delays not only limit the usability of OBS networks for transporting delay-sensitive traffic, but also impair TCP throughput performance.[17] In this section, we argue that the burst length adopted by a trunk should reflect its traffic volume so as to minimize the burst assembly delay, then introduce a proactive reservation scheme that leverages knowledge of the trunk's burst size and the predictability of the transmission schedule of regulated traffic to hide the transmission offset latency.

The average burst size used in the network is dictated by bandwidth efficiency considerations. A burst is typically formed of multiple IP packets. Packets are buffered at the burst-assembly stage until either the desired burst size is met or a limit on packet waiting time in that stage is violated, in which case, padding is used to achieve the desired burst length. The burst assembly time is a concern only in trunks with low arrival rates, or in case the ingress lacks adequate buffering, thus resulting in excessive packet loss. For trunks with low packet arrival rate, the burst assembly latency can be improved by adopting a burst length and a waiting-time limit that are commensurate with the arrival rate.

The transmission offset is the difference between the time of transmission of data burst and that of the corresponding reservation request. To avoid burst dropping, the offset must be large enough to allow for the queuing and processing (header processing, forwarding, and resource scheduling) delays faced by the request packet along the transmission path, and hence must be proportional to the number of hops.[8] In order to maintain efficient use of link bandwidth under TCP traffic, core routers in packet-switched networks should have a bandwidth-delay product worth of packet buffers at each interface. Similar reasoning applies to the control path in OBS switches to minimize the chance of a reservation packets being lost due to temporary congestion. Taking into consideration the average number of packets per burst and the difference in packet size between the IP traffic and control traffic, the queuing delay experienced at each hop by a reservation packet can be in the millisecond range in periods of transient congestion.

In principle, the reservation process for a burst transmission can be initiated proactively–before the formation of the burst to hide the transmission offset latency. The main concern is the bandwidth waste due to unused reservations. Given an efficient bandwidth allocation, that is, no trunk is allocated more bandwidth than it demands, then the trunk's burst queue at the ingress constitutes a single server FIFO queue with bursty arrivals (burst formation inherits the burstiness of packet arrivals) and burst departures occur at the instants for which there are reservations. A fundamental result of queuing theory is that for a queuing system with general arrival and service processes, as the arrival rate approaches the service rate at steady state (utilization approaches 1), the probability of the system being idle approaches 0,[18] that is, no wasted reservations. This queuing model assumes a infinite buffer capacity; in practice, the buffer must be large enough to accommodate the bursty arrivals. As mentioned above, for TCP traffic, the buffer size must accommodate a bandwidth-delay product worth of packets (bursts) to achieve efficient link (channel) utilization. We conclude that given efficient allocation of network bandwidth, and adequate burst buffer capacity at the ingress, bandwidth waste due to unused reservations is not a concern. This is corroborated by simulation results in Section 5.

We propose a periodic proactive reservation scheme that combines periodic shaping with proactive reservation. Periodic shaping can be viewed as the generation of time-stamped tokens; the difference between the time-stamps of two consecutive tokens is equal to the trunk's period, the reciprocal of its bandwidth allocation in $\mathrm{burst/s}$. The burst at the head of the queue is released only when the clock time becomes equal to the time-stamp for the earliest generated tokens. Consumed and expired tokens are destroyed. The process of generating a token is the process of initiating the reservation process (generating a request packet) for burst transmission after a certain offset. This process occurs periodically at a rate equal to the trunk's bandwidth allocation. We refer to the transmission offset as the reservation *horizon*.[*] As is the case with a regular OBS transmission offset, the horizon must be an upper bound on the queuing and processing delay witnessed by the reservation packet along the path. However, this latency is totally hidden from the burst as illustrated in Figure 1. Bursts suffer shaping delays in the burst queue. These delays are equivalent to those suffered in a packet-switched network with ingress shaping. Note also that in OBS, limited buffering at the core routers guarantees small queuing delay beyond the ingress.

To minimize the offset delay, Turner[5] and Xiong and associates[8] proposed buffering the data bursts at the core routers to maintain the difference between the arrival time of the reservation request and the corresponding burst at downstream nodes. The merit of this approach is that the delay incurred by a packets within a burst reflects the actual delay faced by

[*] We make the distinction for reasons that will become apparent in the next section.

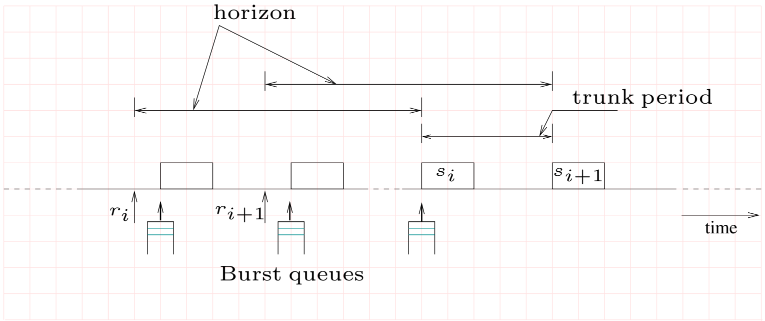


Figure 1. Periodic proactive reservation. The reservation process from the ingress viewpoint: Reservation request r_i yields token s_i. When a token is available, the burst at the head of the queue is transmitted.

16	32	32	32	8	8
trunk ID	arrival time	departure time	offset	channel	shape

Figure 2. Reservation request format. Field sizes are shown in bits.

the corresponding request packet along the control path. Its disadvantage is that it introduces synchronization requirements between the data path and the control path within the routers. In addition, packets within the data burst still incur the scheduling and forwarding delay at each node. A recently proposed scheme named Forward Resource Reservation (FRR)[19] also attempts to partially hide the offset latency by starting the reservation process for a burst transmission starts with the arrival of the burst's first packet at the ingress. FRR does not benefit from traffic aggregation, shaping, or knowledge of the burst size. The main goal of FRR is the accurate prediction of the burst size and the burst assembly delay after the arrival of the initial packet.

3.2. Proactive reservation in PRS

PRS uses the periodic proactive scheme introduced in Section 3.1 above, which can be classified as a Just Enough Time (JET) reservation scheme.[3] Figure 2 shows a possible format for the reservation request message. The `arrival time` and `departure time` fields are used to account for the time spent by the reservation packet in each router as described in Ref. 3. The `channel` field identifies the wavelength channel on which the burst will arrive. The `shape` field is used by the scheduler. In case the trunk allocation is at or below 50% of channel bandwidth, the `shape` field is interpreted based on its most significant bit, which is set only if the trunk's bandwidth allocation is less than 50% of a wavelength channel bandwidth; `shape` then provides the desired length of time between successive bursts in units of trunk's burst size. If the allocation is greater than half the channel bandwidth, then `shape` is interpreted as the maximum number of back-to-back bursts from the same trunk before a gap of trunk's burst size should be forced by the scheduler. The `shape` field is included in every reservation to allow for dynamic bandwidth allocation.

The use of proactive reservation offers a simple yet effective way of improving the efficiency of the control channels by bundling the reservation requests for a trunk over an *object interval* in a single reservation packet. This is especially desirable when the network has an IP-based control plane. In such settings, a reservation protocol data unit is encapsulated in IP datagram, which is in turn augmented with a shim-header before being encapsulated in a layer-2 frame for transmission. In case each IP datagram carries only one reservation request, the efficiency of the control channel is below 50% assuming 20-Bytes IP headers and ignoring the shim-header and layer-2 header overheads. In addition, note that only the `offset` and the `channel` fields need to be repeated for each reservation within the request packet. The larger the length of the object interval the greater the improvement in control channel efficiency. However, the length of the largest offset in a reservation packet dictates the size of the guard bands around the data bursts, which directly affects the efficiency of the data channels. Given that a single reservation request in each datagram can be considered a special case, we describe the reservation process in terms of the more general case.

Figure 3 shows an example where a reservation request packet, R, carries n scheduling requests $r_1, r_2, \ldots, r_n$ resulting in reservations s_1 through s_n at the ingress router. Periodically the switch controller at the ingress router initiates the

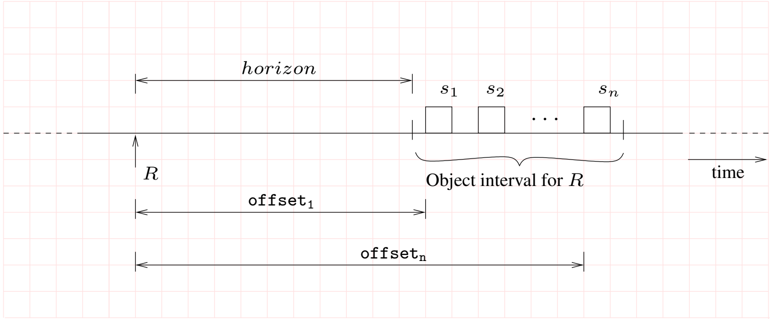


Figure 3. Trunk reservation requests over an object interval bundeled into one request packet R.

reservation process to fulfill the bandwidth allocation for the trunks originating at that node within a future *object interval*. The time between transmitting consecutive request packets (the reservation cycle) for a trunk is equal to the length of the object interval so that new reservations are made at the same rate they are consumed. Suppose the length of the object interval is o, and consider a trunk with bandwidth allocation β, and burst size B. The number of burst transmissions required to fulfill the bandwidth demand within an object interval of length o are $n = o * \beta / B$, assuming that the o is such that $o * b$ is an integral multiple of B. For a 40Mb/s trunk and 4Kb bursts, a 4ms object interval contains 40 burst transmissions. Referring once again to Figure 2, the reservation request packet would contain 40 `<offset,channel>` pairs in addition to the common headers. Moreover, the 20-Bytes IP headers overhead becomes relatively small compared to the reservation packet payload of 680 Bytes.

After locally scheduling the requested transmissions, a reservation request packet is created and forwarded on the control channel to the next switch along the trunk's path where once again the switch controller attempts to schedule the requested transmissions; the process is repeated until the request packet reaches the egress node where it is discarded. Unsuccessfully scheduled requests are marked as *erased* by setting their `offset` fields to zero in the forwarded request packet. If all requests were unsuccessfully scheduled at a core router, the reservation packet is dropped.

4. TRUNK SHAPING AND SCHEDULING

TCP interprets packet loss as a congestion signal and reacts by reducing its sending rate. Therefore, a challenging problem for OBS network designers in an IP backbone is the trade-off of burst loss rate (which translates to packet loss rate) to network utilization, given very limited buffering at the core switches. On-demand OBS reservation schemes propagate the IP traffic burstiness to the network's core. Independently from load, accidental synchronization among bursty trunks leads to high loss rate when the buffer capacity is small. An advantage of packet-switched networks over OBS is that through large buffering capacity, packet switches are capable of sustaining bursty traffic at high-level of link utilization without excessive packet loss.

PRS regulates trunks into periodic streams at the ingress routers to approximate the utilization/loss performance of discrete-time queues with periodic arrival streams – which were studied as models for ATM multiplexers,[20] and are known to allow high bandwidth utilization with modest buffer capacity requirements. One particular model, the $nD/D/1$ queue[21] assumes fixed-size cells and time-slotted output link, with a slot duration equal to cell transmission time. The traffic consists of n periodic streams having independent but equal periods of length M slots. Each of the n streams chooses a slot uniformly at random from 0 to $M - 1$ where it produces a packet each period. Packets arriving to the queue are served in FIFO manner. The $nD/D/1$ queue has some structural similarities with OBS: the choice of the slot (i.e., packet holding at the source until a specific slot), despite not being conveyed to the multiplexer ahead of packet arrival, is analogous to reservation in a slotted OBS system with one output channel, and the FIFO queue is analogous to a first-fit burst scheduling algorithm.

A plot of the complementary queue length distribution for the $nD/D/1$ at 90% utilization is shown in Figure 4. This distribution is an upper bound on the blocking probability of a finite $nD/D/1$ queue. The $nD/D/1$ model applies to streams of equal bandwidth allocations. Upper bounds on queue length distribution for the case of different bandwidth allocations are provided in Ref. 20 where it was observed that the worst buffer requirement occurs when all trunks have

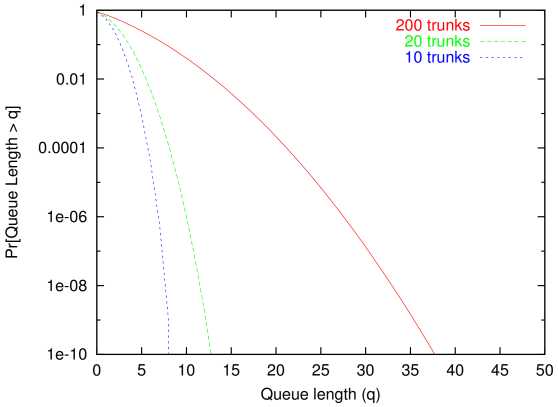


Figure 4. Upper bound on blocking probability of a finite $nD/D/1$ queue at 90% utilization.

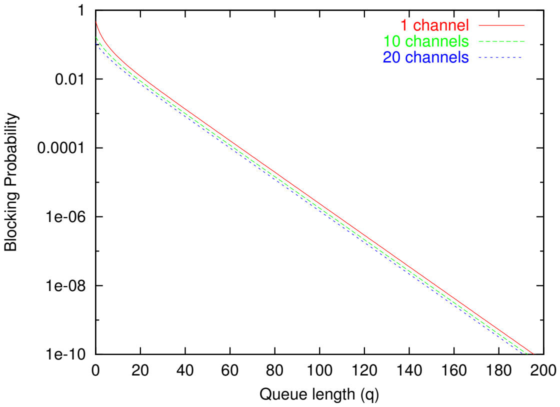


Figure 5. Blocking probability of an $M/M/m/L$ system at 90% utilization. At high loads, increasing the number of servers (channels) does not improve the blocking probability.

the smallest allocation (i.e., largest number of trunks). Thus the case of equal bandwidth allocation should be used in determining the buffer capacity requirements.

In a network setting, a trunk traverses multiple hops, and bursts encounter variable buffering delay at each hop. The variability in delay between successive bursts destroys the periodicity of the trunk. Consider the case of a variable delay with a bound equal to the trunk's period. After traversing h hops, the trunk may have groups of h back-to-back bursts. To avoid excessive loss at downstream switches, the scheduler at the core routers in PRS need to maintain the shape of the trunks through scheduling constraints.

Chaskar and associates[22, 23] recognized the need for shaping at the ingress of OBS networks and proposed enforcing an exponentially-distributed inter-arrival time between consecutive transmission requests from the same trunk in order to obtain the blocking performance of a bufferless m-server-loss system ($M/M/m/m$) at each core router. They observed, that Poisson shaping leads to a substantial improvement over unregulated resource allocation at light load.[23] The authors also provide a traffic engineering framework centered around Erlang's loss formula[24] for trading-off the utilization of network links to loss probability given the lack of buffers at the network's core. The lack of buffers obviates the need for shaping at the network's core.

A more general model for Poisson shaping is that of an m-server system with finite buffer ($M/M/m/L$). The plot in Figure 5 shows the blocking probability at 90% utilization of an m-server system with a bound L on the number of customers allowed in the system at any time and Poisson arrival and service processes–the $M/M/m/L$ queuing system.[24] At high load, a system with Markovian arrivals and service times has a large buffering requirement that is almost independent of the number of channels. The variability in packet arrival process at a backbone router is known to be much higher than that of Markovian arrivals suggesting that even larger buffering capacity would be required in practice. By comparing figures 4 and 5, it is clear that periodic shaping yields better loss/utilization performance than Poisson. The $nD/D/1$ queue requires a buffer capacity of less than 40 to support a loss rate of 10^{-10} for 200 trunks at 90% load, whereas the Markovian queue requires a buffer capacity of 200. Note that the number of trunks plays no role with Poisson arrivals since a mix of a set of Poisson processes is also a Poisson process.

Reacting to phase synchronization

It can be shown that given utilization ≤ 1, a link subject to periodic reservations implements a guaranteed rate server for each stream with latency no greater than its period. The complementary distribution in Figure 4 indicates that there is a large gap between typical behavior and worst case guarantee. This suggests that to avoid worst case buffering and waiting, switches should detect trunks experiencing high blocking rate and signal the ingress to randomize its phase with respect to competing trunks. Phase synchronization may result in severe throughput deterioration for affected trunks. In order to guarantee robustness to phase synchronization, switch controllers can maintain a moving average of the fraction of blocked

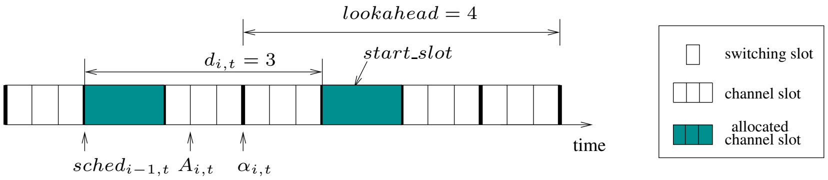


Figure 6. Calendar scheduling example.

reservations from each trunk that passes through them. If that fraction increases beyond a predefined threshold, it indicates that there is phase synchronization. A Phase Synchronization Indicator (PSI) packet containing the trunk identifier is sent back to the ingress. Upon receiving a PSI for a particular trunk, the controller at the ingress randomizes the phase of the request generation cycle for that trunk. Provisions to avoid congesting the control channels with PSI packets, such as a limit on the rate of generation of PSIs by individual switches, must be employed to avoid the congestive collapse of the control channels in the case some wavelength channels get oversubscribed due to faulty ingress controllers.

Scheduling in PRS

In PRS, the scheduling algorithm is deployed at the output interfaces of the ingress and core routers to process reservations and allocate channel, and if necessary, buffer resources to incoming bursts. The scheduler provides a best-effort service in the sense that it may fail to schedule a subset of the requests it receives due to channel and buffer constraints, in which case the request is blocked, as described in Section3.2. The scheduler also acts as a shaper by delaying individual bursts to maintain the temporal distance between successive bursts from the same trunk.

The switching elements within core OBS routers operate in discrete time slots smaller than the burst duration. We shall refer to these as *switching slots* with duration s nanoseconds. Since OBS-based networks are not globally synchronized, requested transmission start time for a burst may not be aligned with switching-slot boundary at the switch processing the reservation request. As described in Ref. 8, guard bands around the burst guarantee that differences smaller than the switch slot size do not affect proper burst reception.

In order to emulate the discrete-time queuing model in Section 4, we require that channels are allocated in slots of duration equal to the burst size. We shall refer to these as *channel slots*, with duration S switching slots. Bursts arriving within a channel slot need to be buffered for a duration that may be a fraction of the burst size, buffer allocation is therefore performed at the granularity of switching slots. We assume that a buffer consists of a set of FDLs capable of buffering from 1 switching slot up to a multiple W of the channel slot duration. The $lookahead = W - 1$ is a bound on the buffering time a burst may experience at the switch in units of channel slots after alignment.

In describing the burst scheduling algorithm, we represent the schedule of a channel or buffer resource as an array (a calendar) where each entry describes the availability of the resource during a time slot of the proper granularity. Consider a channel calendar, each entry corresponds to a slot of duration equal to that of the burst size. Let $a_{i,t}$ be the requested arrival time for burst i from trunk t in units of absolute time. The alignment constraint requires that the burst be allocated a buffer starting at switching slot $A_{i,t} = \lfloor a_{i,t}/s \rfloor$ for a number of switching slots equal to $b = S - (A_{i,t} \bmod S)$. The aligned arrival time of the burst is $\alpha_{i,t} = A_{i,t} + b$.

The shaping function of the scheduler involves temporally spacing successive bursts from the same trunk. The distance that the scheduler enforces between the $i-1$th and ith bursts from trunk t is denoted by $d_{i,t}$ where $0 \leq d_{i,t} \leq lookahead$. This value is set based on the value of the `shape` field in the request for burst i.

The example in Figure 6 illustrates the working of the scheduling algorithm. It shows the calendar for one channel, where channel slots are drawn with thick boundaries. For the purpose of this example, a channel slot consists of only three switching slots. Due to the shaping distance constraint $d_{i,t}$—three channel slots from the preceding channel allocation for the same trunk $sched_{i-1,t}$—burst i can only be scheduled starting at the channel slot marked $start\_slot$, but not beyond the $lookahead$. The slot at $start\_slot$ is already allocated, but the next one is available and would therefore be allocated to burst i. The scheduler may attempt scheduling the burst across multiple wavelengths, in which case the calendars for the candidate wavelengths may be checked in parallel. The channel slots within a calendar may be checked sequentially or in parallel using an associative structure. Note that the response time of the scheduler in the case of sequential implementation is determined by the $lookahead$, which is typically small because of the limited buffer capacity.

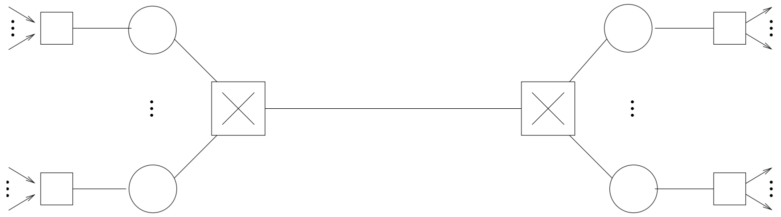


Figure 7. Single-bottleneck topology with n forward and n reverse trunks

5. PERFORMANCE EVALUATION

To evaluate the performance of the proposed traffic shaping and scheduling schemes under TCP traffic, we implemented a PRS simulator in ns.[25] We rely on the sum of throughputs of TCP flows crossing a link to evaluate data channel utilization performance given a bounded scheduling lookahead. As efficient utilization of the individual channels is the motivation behind the proposed shaping and scheduling schemes, we consider only the case of one data channel per link in addition to a control channel. TCP traffic is generated by long-lived FTP sessions. Though most flows in the Internet are short-lived, 80% of the traffic belongs to long-lived TCP sessions. Therefore, when it comes to the evaluation of network throughput performance, we are primarily interested in the steady-state throughput of long-lived TCP flows.

5.1. The single-bottleneck case

In Section 4, the number of competing trunks was the main factor affecting the queue length and waiting time in $nD/D/1$ queues. In this section, we evaluate the performance of PRS by varying this parameter. Figure 7 shows the topology used to conduct the experiments. A forward trunk is composed of 20 FTP flows. Each FTP source is connected to the trunk's edge node which, in turn, is connected to the PRS router. We choose to not let trunks share edge nodes to ensure that the PRS link is the only bottleneck in the system. Similarly, the FTP sinks for a particular trunk are connected to its egress edge, which is in turn connected to a PRS router. A reverse trunk from the egress to ingress carries the TCP acknowledgments for the forward traffic. All links have 1 ms propagation delay and 20 Mb/s capacity.

We chose to use scaled-down channel capacities, trunks and number of flows per trunk for simulations to complete in reasonable amounts of time. This does not affect the conclusion we reach as we evaluate the performance of PRS under TCP traffic. The TCP segment size is set to 256 Bytes, burst size is 512 Bytes. The reservation horizon is 10 ms and the object interval length is 10 ms. Note that the bandwidth-delay product is equal to 80 packets, which is the buffer requirements for the bottleneck router with unregulated TCP traffic to achieve full bandwidth utilization. We fix edge router buffer size at 100 packets to allow smoothing without excessive loss. Since all trunks have equal number of flows that traverse identical paths, all trunks have equal allocations of the shared bottleneck bandwidth. Each trunk initiates its reservation process by choosing a random time instant during the first second of simulation time. FTP flows start at random moments during the next second. Experiments reported here were run at channel load of 90%. Subsequent simulations indicate that increasing the load up to 99% does not affect the utilization/loss performance when periodic traffic shaping and burst alignment are used.

5.1.1. Effects of shaping and alignment

We begin by demonstrating the benefits of alignment and periodic shaping. We let the number of trunks vary from 1 to 10 trunks and evaluate the bottleneck utilization performance with a fixed lookahead of 5 bursts. Note that the throughput of TCP flows and hence link utilization is an indicator of loss performance. Figure 8 compares the throughput achieved using Poisson shaping and periodic shaping with alignment. The combination of alignment and periodic shaping result in double the utilization achieved by Poisson shaping (92%). Enforcing the alignment constraint on Poisson-shaped traffic result in significant improvement in throughput from 40% to 73%.

5.1.2. Effect of number of trunks

Next, we demonstrate the effect of increasing the number trunks competing at the bottleneck on buffer requirement for achieving high link bandwidth utilization. In Figure 9, the normalized throughput and the loss rate are plotted against the lookahead for different number of trunks. The normalized throughput is the ratio between the sum of the throughputs of trunks crossing the bottlenecks divided by the utilizable channel capacity (18Mb/s). Note that the loss rate beyond

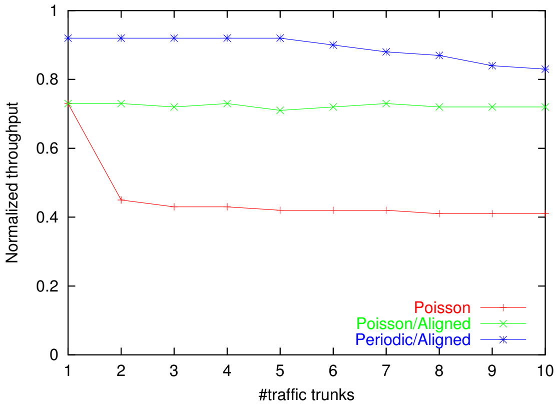
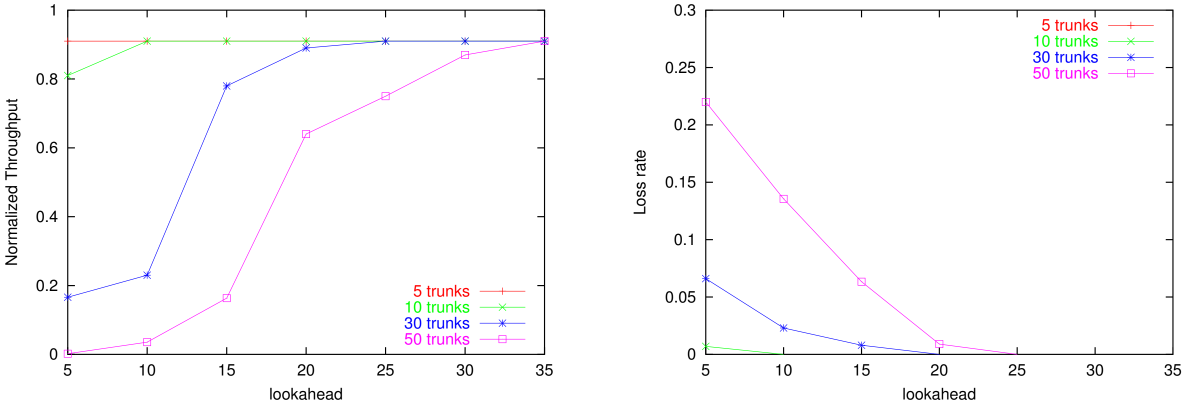


Figure 8. The benefits of periodic shaping and burst alignment.



(a) Bandwidth utilization.

(b) Loss rate.

Figure 9. Effect of number of trunks on bandwidth utilization and loss rate.

$lookahead = 35$ is zero in all curves. We conclude that the required lookahead value grows slower than the number of trunks. Note that the lookahead is an upper bound on the number of bursts simultaneously waiting in the buffer.

5.2. The case of multiple bottlenecks

In case of multiple bottlenecks, as data bursts are subjected to a variable yet bounded waiting time delay at each hop, we expect the bandwidth utilization and buffer occupancy to deteriorate due to burstiness compared to the single-bottleneck performance. To emphasize the effect of burstiness, we designed the topology in Figure 10. Each link is shared among exactly 10 trunks. A subset of the trunks continue while others exit after crossing only one bottleneck. Continuing trunks are subjected to competition on each link to increase their burstiness. Figure 11 shows that a lookahead of 5 bursts is no longer sufficient to maintain low loss rate for the traffic after crossing multiple bottlenecks. This is an indication that in networks with large diameter, traffic shaping only at the edge is not sufficient. Figure 11-b shows the benefit of shaping at the core routers. There is a trade-off between buffering opportunities and shaping. Given a lookahead of 5 channel slots, if the scheduler enforces a minimum distance of 5 channel slots between consecutive burst reservations from the same trunks, the burst has only one transmission opportunity, if the channel is busy during that time, the burst is dropped. For the 5-hop network, the best performance was achieved when the enforced shaping distance was 2 channel slots.

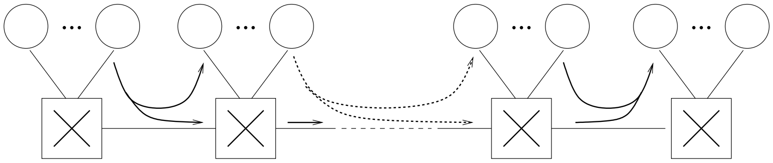
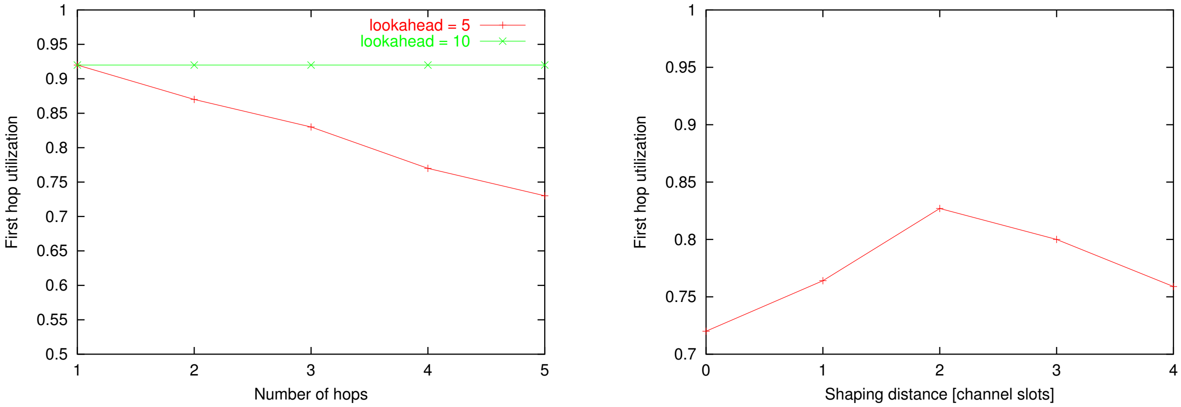


Figure 10. Multi-bottleneck topology.



(a) Bandwidth utilization.

(b) Effect of shaping at the core.

Figure 11. Effect of number of hops on bandwidth utilization, and the role of shaping at the core.

6. CONCLUDING REMARKS

In this paper, we introduce the PRS framework for implementing IP/WDM. This framework modifies OBS in a number of ways: 1) enforcing uniform traffic shaping both at the ingress and at intermediate nodes, 2) discretizing the calendar at each node into slots and aligning burst reservations to minimize calendar fragmentation, 3) applying proactive burst reservation rather than on-demand reservation, and 4) packing multiple burst reservations into one reservation request packet. The first two modifications lead to increased bandwidth and decreased blocking probability, while the third modification reduces the end-to-end delay and the fourth modification reduces the traffic on the control network and allows for a more efficient scheduling. Due to space limitation, we only present in this paper simulation results to show that PRS efficiently utilizes the bandwidth of the optical data channels and reduces the blocking probability to a level that is adequate for supporting IP traffic. Results showing that PRS reduces the end-to-end delay and the control overhead will be presented in subsequent work.

REFERENCES

1. D. J. Blumenthal, J. E. Bowers, L. Rau, H.-F. Chou, S. Rangarajan, W. Wang, and H. N. Poulsen, "Optical signal processing for optical packet switching networks," *IEEE Optical communications* **1**, pp. S23–S29, February 2003.
2. Stanford University, "The Optical Router Project." `http:///klamath.stanford.edu/or/`.
3. M. Yoo, M. Jeong, and C. Qiao, "A high speed protocol for bursty traffic in optical networks," in *SPIE'97 Conf. For All-Optical Networking: Architecture, Control, and Management Issues*, pp. 79–90, 1997.
4. M. Yoo and C. Qiao, "Optical Burst Switching (OBS) – a new paradigm for an optical internet," *Int'l J. High-Speed Networks* **8**(1), 1999.
5. J. S. Turner, "Terabit burst switching," *Int'l J. High-Speed Networks* **8**(1), pp. 3–16, 1999.
6. Y. Chen and J. S. Turner, "WDM burst switching for petabit capacity routers," in *IEEE Milcom*, 1999.
7. L. Tančevski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti, and T. McDermott, "Optical routing of asynchronous, variable length packets," *IEEE JSAC* **18**, pp. 2084–2093, October 2000.

8. Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE journal on selected areas in communications* **18**, pp. 1838–1851, October 2000.
9. J. Xu, C. Qiao, and G. Xu, "Efficient channel scheduling alogrithms in optical burst switched networks," in *IEEE INFOCOM*, 2003. To appear.
10. C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *IEEE Commun.* , September 2000.
11. S. Zheng, Y. Xiong, M. Vandenhout, and H. C. Cankaya, "Hardware design of a channel scheduling algorithm for optical burst switching routers," in *Optical Transmissions and Equipment for WDM Networking, ITCOM*, *Proceedings of SPIE* **4872**, pp. 199–209, 2002.
12. H. T. Kung and S. Wang, "TCP trunking: Design, implementation, and performance," in *IEEE ICNP*, (Toronto, Canada), 1999.
13. D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwith-delay product networks," in *ACM SIGCOMM*, (Pittsburgh, Pennsylvania), 2002.
14. K. Thompson, G. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network* **11**(6), pp. 10–23, 1997.
15. C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *To appear in IEEE Network* , 2003.
16. Sprintlabs. The IP Monitoring Project, "Packet Trace Analysis." `http://ipmon.sprintlabs.com/`.
17. A. Detti and M. Listanti, "Impact of segments aggregation on TCP Reno flows in optical burst switching networks," in *Infocom*, (New York), June 2002.
18. L. Kleinrock, *Queueing Systems, Vol. I: Theory*, Wiley Interscience, Boston, MA, 1974.
19. J. Liu and N. Ansari, "Forward resource reservation for QoS provisioning in OBS systems," in *IEEE Globecom 2002*, pp. 2777–2781, (Taipei, Taiwan), November 2002.
20. J. W. Roberts and J. T. Virtamo, "The superposition of periodic cell arrival streams in an ATM multiplexer," *IEEE Trans. Commun.* **39**, pp. 298–303, Feb. 1991.
21. P. Humblet, A. Bhargava, and M. G. Hluchyj, "Ballot theorems applied to the transient analysis of nD/D/1 queues," *IEEE/ACM Transactions on Networking (TON)* **1**(1), pp. 81–95, 1993.
22. H. M. Chaskar, S. Verma, and R. Ravikanth, "A framework to support IP over WDM using Optical Burst Switching," in *IEEE/ACM/SPIE Optical Networks Workshop*, (Richardson, Texas), January 2000.
23. S. Verma, H. Chaskar, and R. Ravikanth, "Optical Burst Switching: A viable solution for Terabit IP backbone," *IEEE Network* , November/December 2000.
24. D. Gross and C. M. Harris, *Fundamentals of queuing theory*, John Wiley and Sons, Inc., 3rd ed., 1998.
25. S. McCanne and S. Floyd, "ns network simulator." `http://www.isi.edu/nsnam/ns/`.