Decoupling Packet Loss from Blocking in Proactive Reservation-based Switching

Mahmoud Elhaddad, Rami Melhem, Taieb Znati
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
{elhaddad, melhem, znati}@cs.pitt.edu

Abstract

We consider the maximization of network throughput in buffer-constrained optical networks using aggregate bandwidth allocation and reservation-based transmission control. Assuming that all flows are subject to loss-based TCP congestion control, we quantify the effects of buffer capacity constraints on bandwidth utilization efficiency through contention-induced packet loss. The analysis shows that the ability of TCP flows to efficiently utilize successful reservations is highly sensitive to the available buffer capacity. Maximizing the bandwidth utilization efficiency under buffer capacity constraints thus requires decoupling packet loss from contention-induced blocking of transmission requests. We describe a confirmed (two-way) reservation scheme that eliminates contention-induced loss, so that no packets are dropped at the network's core, and loss can be incurred only at the adequately buffer-provisioned ingress routers, where it is exclusively congestion-induced. For the confirmed signaling scheme, analytical and simulation results indicate that TCP aggregates are able to efficiently utilize the successful reservations independently of buffer constraints.

1. Introduction

Optical Burst Switching (OBS) [20, 23, 24] is a reservation-based switching technology that is being considered as an access control and switching solution for IP-over-WDM networks. 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 over dedicated control channels (control wavelengths) along the path of the burst. The header packets are processed electronically at each switch to determine the next-hop, and to reserve switch resources (e.g., output channel, buffer, and wavelength converters) to the corresponding bursts. The resource allocations at a given switch form a schedule according to which arriving bursts transparently cut through the switch. 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.

Motivated by the dominance of TCP traffic on the Internet and the sensitivity of TCP throughput to packet loss, we introduced Proactive Reservation-based Switching (PRS) [5], a reservation-based slotted switching architecture that is built on the switching and signaling technology of OBS, to achieve the following goals: (1) To minimize the buffer capacity required at the core switches for TCP traffic to utilize the network bandwidth efficiently, (2) to allow flexibility in the allocation of bandwidth to elastic ingress–egress traffic aggregates, (3) to minimize the delay suffered by packets at the ingress of the network, and (4) to promote efficient use of control plane bandwidth. PRS defines a set of high-level transmission control functions for IP networks based on the OBS technology, specifically, reservation signaling, traffic shaping and scheduling at the edge routers and core switches.

PRS exploits the persistence of ingress–egress IP traffic aggregates (trunks) [19] to minimize the packet waiting time at the ingress router by having the ingress routers proactively generate reservation requests on behalf of the trunks. This is in contrast to on-demand reservation in OBS, where the arrival of a burst of packets triggers the generation of such requests. Proactive reservation also enables the use of confirmed (two-way) reservation signaling by initiating reservation for a time slot early enough that, if the request is successful, the confirmation is guaranteed to be received at the ingress in time for the data to be released.

PRS reduces contention by regulating the requested time slots by each trunk into a periodic stream, thus minimizing the link buffer capacity needed for loss-sensitive IP flows to achieve high bandwidth utilization of network links. Still, due to limitations of optical buffering using FDLs (fiber delay lines or loops), such buffer requirements may not be cost-effective or practical. In this paper we consider the problem of maximizing the network throughput given buffer capacity constraints and traffic that is subject to loss-based congestion control.

Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)
0-7695-2221-1/04 $ 20.00 IEEE

First, we describe a mathematical model of reservation blocking at a PRS link and use it to obtain an upper-bound on the throughput achievable by a trunk. This bound is the fraction of reservation requests that are successful (i.e., not blocked due to contention), and is a function of the load, the number of competing trunks, and the buffer capacity at the links constituting the trunk's path. It should be used to derive routing constraints during the network operation phase, and in provisioning links with wavelength channels and buffer capacity during the network design phase.

Next, we show that in a PRS network using unconfirmed (one-way) reservation signaling, the ability of TCP trunks (trunks of TCP-fair flows) to utilize their successful reservations is highly sensitive to the available buffer capacity due to blocking-induced packet loss. Blocking-induced loss is eliminated if confirmed reservation signaling is used. We show that in this case, TCP trunks are guaranteed to achieve high utilization of the successful reservations. We corroborate these results using TCP-driven `ns-2` simulation. Simulation is also used to study the effects of the frame (burst) size and the round-trip delay on the bandwidth efficiency in addition to the performance of web-like transfers under confirmed and unconfirmed signaling.

In the next section, we provide an overview of PRS. In Section 3, we introduce a mathematical model for blocking at a PRS link, then in Section 4, we develop the bound on the rate of successful reservations. Section 5 contains the analyses of TCP throughput under unconfirmed and confirmed reservation signaling, and Section 6 presents the simulation results. Concluding remarks are presented in Section 7.

2. Proactive Reservation-based Switching

PRS is a reservation-based switching architecture based on the technology of OBS. It employs a reservation protocol based on Just Enough Time (JET) signaling [23], and uses dedicated control wavelengths along each link to carry reservation protocol traffic. In this section, we describe the network capabilities required to support PRS—namely, traffic aggregation, routing, and aggregate bandwidth allocation—then present an overview of its transmission control functions.

2.1. Traffic aggregation, routing, and bandwidth allocation

In order to facilitate traffic aggregation at the ingress nodes, PRS is based on a label-switched control architecture. Ingress traffic is aggregated into trunks, based on the egress address, and possibly, quality of service requirements. Traffic aggregation allows for bandwidth-efficient reservation signaling 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 frame scheduling based on trunk membership.

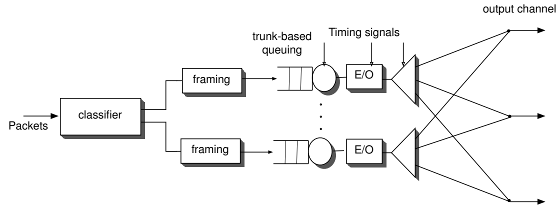

Figure 1. Ingress data-path organization.

A Routing and Wavelength Assignment (RWA) algorithm (see [14] and references therein) is used to compute for each trunk a path, and a set of wavelengths that it may use on each link along that path. The algorithm is run whenever there is a change in the network topology or long-term trunk demands.

By regularly spacing frame transmissions, the ingress routers regulate trunks into periodic streams, so that contention at the core is minimized, and TCP trunks become able to efficiently utilize network bandwidth without large core buffer capacity requirements. Since the ingress routers are electronic packet switches, large buffers are assumed to smooth the bursty packet arrival process without incurring substantial packet loss. The architecture of the edge router ingress data-path (the path from an input IP interface to an output PRS interface) is depicted in Figure 1. For one-way signaling, the term *schedule* used in the figure refers to attempted reservations, as opposed to a schedule of confirmed reservations in the case of two-way signaling.

Under an efficient network-wide allocation of bandwidth, no trunk is allocated more bandwidth than it demands. To maintain efficiency despite trunk shaping at the edges, the bandwidth allocations need to be dynamically revised. Techniques for the estimation of fair and efficient bandwidth allocations at the level of traffic aggregates were proposed in [8, 11, 12].

2.2. Proactive reservation

In order to minimize the delay experienced by packets at the ingress routers, PRS exploits the periodicity and persistence of shaped aggregates [19] by having the ingress of each trunk initiate channel reservations along the trunk's path without waiting for corresponding frames to form (or start forming). Since under JET signaling, a channel reservation request must specify the reservation start-time and end-time, proactive reservation requires knowledge of the frame duration before it is formed. PRS is time slotted: it assumes a universal frame size, for example in bits, so that each switch can map it locally to a number of time slots (clock ticks or *switching slots*) that is used locally as the minimum allocation unit of wavelength channels. This unit is henceforth called a *channel time slot*, or simply, channel slot.

Proactive reservation should be contrasted with on-demand reservation signaling used in OBS. A fundamental assumption in the on-demand scheme is that users initiate one-time data transfers whose durations (the burst durations) are unpredictable but are generally too small with respect to the round-trip delay required to setup a circuit [21, 23]. The data is buffered at the ingress of the OBS network for the duration of the offset needed for the request to be processed at each switch along the route before the arrival of the burst. In a transport network environment, on-demand reservation is likely to cause transient congestion in the control plane, in which case, reservation packets will face large queuing delays. The offset latency suffered by the packets at the ingress must be large enough to accommodate the worst-case request packet delay along the control path. In contrast, proactive reservation eliminates the offset latency (though not the offset) by reserving time slots within a time interval before it begins. Trunk packets arriving at the ingress during that interval are packed into frames and transmitted according to the schedule of existing reservations.

Proactive reservation signaling promotes control plane efficiency by dividing the overhead of a reservation packet over multiple reservation requests [5]. The ingress periodically generates a control packet that contains the requested time slots during a target interval. The number of requests within an interval reflects the trunk's bandwidth allocation. Proactive reservation over a target interval can be thought of as setting up an ephemeral TDM channel that tolerates partial blocking, and is periodically renewed to adapt to the trunk's changing bandwidth allocation.

By eliminating the offset latency, a proactive reservation scheme allows confirmed (two-way) reservation signaling. In two-way signaling, the control packet has an additional field identifying the packet type as either a request, or a confirmation packet. For each request packet the egress receives, it generates a confirmation packet with the same content except for the trunk identifier, which is set to that of the corresponding trunk along the reverse path. The confirmation packet is forwarded along the reverse path, and its request offsets are adjusted at the switches to account for the propagation and processing delays. In one-way signaling, records for blocked requests may be removed from the reservation packet or simply marked before the packet is forwarded to the next switch along the trunk's path. This is also true for confirmed signaling except for schemes supporting resource reclamation on the reverse path. In such case, blocked reservations should only be marked so that the channel and buffer resources they hold are released. In this paper, we do not consider resource reclamation.

The large offset delay due to queuing in the control plane, bundling of multiple time-slot requests into a single request packet, and/or two-way signaling implies larger drift among the clocks along a trunk's path. In a network based on JET signaling, clock drift is countered by wrapping frames with guard intervals larger than the worst expected drift. In the case of slotted channels, the slot duration is augmented by the length of the guard intervals. Therefore the frame duration needs to chosen so as to minimize their overhead [17, 22]. An alternative solution to the drift problem is equipping edge and switching nodes with GPS as a source for high precision timing [2, 4], in which case, the length of the guard intervals becomes independent from the offset.

2.3. Shaping and contention resolution

The architecture of a PRS switch is similar to those proposed for OBS (for example in [17, 22]). Logically, switches are output buffered with each output wavelength having a dedicated buffer of at least one frame place (for the purpose of frame alignment). Requests for slot reservation on an output channel are processed by the channel scheduler. The scheduler may decide to buffer the corresponding frame upon arrival, if necessary, to align it with channel time-slot boundary. The scheduler may also buffer the incoming frame to resolve contention, or to maintain the periodicity of trunk traffic according to a field in the request packet where the ingress specifies a shaping constraint. For example, the desired number of slots between consecutive frames. Wavelength conversion may be used, together with buffering, for contention resolution. In order to isolate the effects of buffer constraints, we focus our attention in this paper on bandwidth utilization in networks without wavelength conversion.

Buffering may be implemented in optics or electronics. The model assumed for optical buffers is shown in Figure 2. A buffer position is implemented using an optical delay loop. Loop delay is the duration of one channel slot. A delay of more than a channel slot is achieved by recirculating the frame through the delay loop. The optical cross-bar is implemented using solid-state optical switches. Frame alignment requires buffering for a fraction of the channel slot duration. This is achieved using a switched Fiber Delay Line (FDL) as shown. See [10] for details on the design of optical buffers using switched delay lines.

A channel scheduler in PRS maintains a time-slot allocation vector (calendar), for each buffer place, as well as for the channel itself [5]. At any given time, the channel scheduler and the switch controller operate on different parts of the calendar. At the beginning of each time slot, the controller configures the channel's data-path (for example, which buffer place is connected to the channel) according to the corresponding entry in the calendar. Calendars have to extend for the duration of largest permissible offset. Assuming a 10μs time-slot duration, a maximum offset of 100 ms requires a calendar size in the order of 10 Kb. Note that this is a high-level description of the scheduling function, which does not preclude more space- or time-efficient calendar implementations than linear arrays.

Channel schedulers perform a lookahead scheduling algorithm to allocate channel and buffer slots to incoming

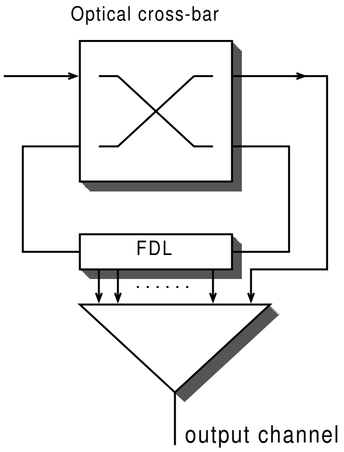

Figure 2. Optical buffer model: A single buffer position.

frames. Starting with the first candidate slot after applying the shaping constraint, the scheduler linearly scans the channel calendar for the first available channel slot, and allocates it. If needed, the scheduler then allocates the necessary buffer slots and wavelength converters.

Under any buffer implementation (whether electronic or optical) scheduling is subject to buffer capacity constraints. Due to distortion and signal decay, optical buffering using recirculating delay loops may place a constraint on the number of channel slots a frame can spend in the buffer. We call this type of constraints a *lookahead constraint*. A maximum lookahead of n slots limits the effective buffer capacity by guaranteeing that no more than n frames will be simultaneously buffered. In addition, a request may become blocked even if one or more buffer places are available, but the time slots within the lookahead are allocated. In this sense, a lookahead constraint is stronger than a buffer capacity constraint. In this paper, we build models of TCP behavior across PRS links assuming only buffer capacity constraints, and use simulation to evaluate the effect of adding lookahead constraints.

3. PRS link model

Consider the single-hop PRS network depicted in Figure 3, where traffic from the ingress nodes is multiplexed at an output channel (link) of the upstream switch.[1] Suppose exactly one trunk originates (terminates) at each ingress (egress) router, and that the total number of trunks is K. Since at a given load, contention increases with the number of competing sources, we will be concerned only with the case where all trunks have identical bandwidth allocations. Slots requested by any given trunk are generated at the ingress as a periodic stream. Suppose that every trunk requests one slot every M slots, then the load on the link is $\rho = K/M \leq 1$ and we say that every M consecutive slots constitute a link *cycle*. Each trunk is equally likely to choose a specific slot and the choice remains fixed from one cycle to the next.

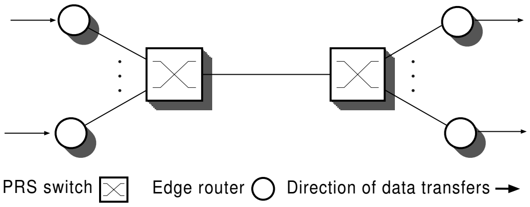

Figure 3. A single-hop PRS network.

Let the link buffer size be q frames. Here, we consider the case where a request is blocked only if the requested slot has been allocated to a competing trunk and no buffer place is available to store the corresponding frame until the next available slot (buffer capacity constraint). Observe that a PRS link provisioned with an infinite buffer has the same buffer occupancy distribution as an nD/D/1 (deterministic) queue [9, 18] having the same utilization and number of periodic sources (trunks).[2] The link blocking probability $\beta(K, \rho, q)$ is thus bounded above by the complementary (tail) queue size distribution of the nD/D/1 queue. Approximating the blocking probability at buffer size q by the complementary distribution ($\Pr[Q > q]$), we get [18]

$$\beta(K,\rho,q) \approx \sum_{r=1}^{K-q} \frac{M-K+q}{M-r}\binom{K}{q+r} \cdot\left(\frac{r}{M}\right)^{q+r}\left(1-\frac{r}{M}\right)^{K-q-r} \quad q \geq 0. \quad (1)$$

This approximation is valid under light and moderate load. Since $\Pr[Q > q]$ is the proportion of time the queue length remains above q, the disparity between the upper-bound and the actual blocking probability increases rapidly beyond 90% load. Note that the probability of the event $\{Q > K\}$ is always zero.

Next, we use this model to characterize the maximum throughput achievable in a PRS network.

4. A bound on the maximum bandwidth utilization efficiency in PRS networks

The maximum bandwidth utilization of a PRS link is the ratio of successful reservation requests for the trunks routed through the link. Note that the probability of a request being successful is the complement of the trunk's path blocking probability. Now, we develop an expression for the

[1] Since we study a single wavelength channel in isolation (no wavelength conversion), we henceforth refer to a channel as a link.

[2] Because the order in which reservation requests are processed may not be that of the arrival of the corresponding frames, the service order in a PRS link is different from the FCFS discipline assumed for the nD/D/1 queue. Whereas, the service order affects the waiting time distribution, queue occupancy remains the same for both systems.

path blocking probability in PRS using the link model introduced in the previous section, and use it to characterize the bandwidth utilization efficiency.

Consider a PRS network composed of a set $\mathcal{L}$ of links. Let $\mathcal{T}$ be the set of trunks in a PRS network. Each trunk $t \in \mathcal{T}$ is routed along a path p_t connecting its ingress and egress, and has a bandwidth allocation b_t $\mathrm{frames/s}$. Let $l \in \mathcal{L}$ be a link in the network, and let $\mathcal{T}(l)$ denote the set of trunks routed through it. The link utilization is $\rho_l = \sum_{t \in \mathcal{T}(l)} b_t / C_l$, where C_l is the link's capacity in $\mathrm{frames/s}$.

Define the mean bandwidth utilization efficiency U_t of trunk $t \in \mathcal{T}(l)$ as the ratio of the trunk's throughput to its bandwidth allocation b_t. Then the mean bandwidth utilization efficiency of link l, U_l, can be expressed as

$$U_l = \frac{1}{\rho_l C_l} \sum_{t \in \mathcal{T}(l)} b_t U_t. \tag{2}$$

Blocking occurs at an output of link when multiple trunks contend for a particular slot at that link and there is no buffer capacity available to delay one or more requests until subsequently available slots. Since the number of requests within a cycle cannot exceed the number of cycle slots ($\rho \leq 1$), a blocked request for a trunk implies that a slot will remain unused on every link along the path of the trunk. It follows that the mean bandwidth utilization efficiency of a link can be expressed in terms of the blocking rate along the paths of the trunks traversing it. Specifically, let $PB(p_t)$ denote the blocking probability along the path p_t of a trunk t. If the trunk is able to fully utilize its successful reservations then $U_t = 1 - PB(p_t)$. In this paper we deal with the case where flows within a trunk have unlimited bandwidth demands. However, successful reservation can still be wasted due to the interaction between contention-induced blocking and loss-based TCP congestion control, hence the following bound on link utilization efficiency

$$U_l \leq \frac{1}{\rho_l C_l} \sum_{t \in \mathcal{T}(l)} b_t \cdot [1 - PB(p_t)]. \tag{3}$$

Now, we use this bound to formulate a trade-off between the length of a path (number of hops) and the bandwidth efficiency of a trunk along that path. We assume that shaping is applied at the core switches so that every link can be modeled as an nD/D/1 queue. Let $p_t = (l_1, l_2, \ldots, l_n)$, the blocking probability at link l_i is $\beta(K_i, \rho_i, q_i)$. Then, we can express $PB(p_t)$ as

$$PB(p_t) = 1 - \prod_{i=1}^{n} [1 - \beta(K_i, \rho_i, q_i)]. \tag{4}$$

By considering the case where all links in the network have the same buffer size q, number of trunk K, and utilization ρ, Equation (4) reduces to

$$PB(p_t) = 1 - [1 - \beta(K, \rho, q)]^n. \tag{5}$$

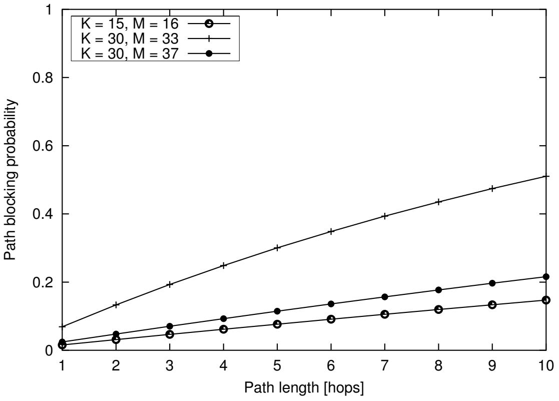

Figure 4. Path blocking probability at $\mathbf{q = 5}$.

If q is taken to be the smallest link buffer size in the network and K the largest number of trunks competing on any link, then, combined with (3), Equation (5) represents the trade-off between these quantities and the distribution of trunk path lengths across a link (hence, network connectivity) on one hand and the desired link bandwidth utilization on the other. Figure 4 shows such a trade-off. It depicts the effect of path length on the blocking probability when the buffer capacity is held at 5 frame places. For example, we can see that to achieve a link utilization close to 86%, trunk routes and the number of wavelengths on the links should be set so that the number of trunks sharing a channel is around 15, with the majority of trunks routed along paths less than five hops in length: At 5 hops and $\rho = 15/16 \approx 0.94$, the success prob. is $1 - 0.08 = 0.92$. Thus the link utilization is $0.94 \times 0.92 \approx 0.87$. By substituting into the equations above, we also find that achieving similar performance at $K = 180, M = 200$ over a 5-hop path requires increasing the buffer capacity only to 12 places. Note that the equations above represent upper bounds on blocking when contention is maximized: trunks competing for slots on an output channel are assumed to be independent, that is, they arrive to the switch on different input interfaces or channels.

5. Performance of TCP trunks in a PRS-based network

In this section, we apply a general model of the throughput behavior of TCP Reno [6] under random loss to analyze the bandwidth efficiency of PRS-based networks using one-way and two-way signaling.

5.1. Unconfirmed signaling

Consider a TCP connection routed along a path having one or more bottlenecks, and let ℓ denote the packet loss probability over the path. Assuming no loss of acknowledgment packets, the TCP throughput-loss formula [15] characterizes the long-term average connection throughput in packets per round-trip time (RTT) as $\frac{3}{2\sqrt{\ell}}$.

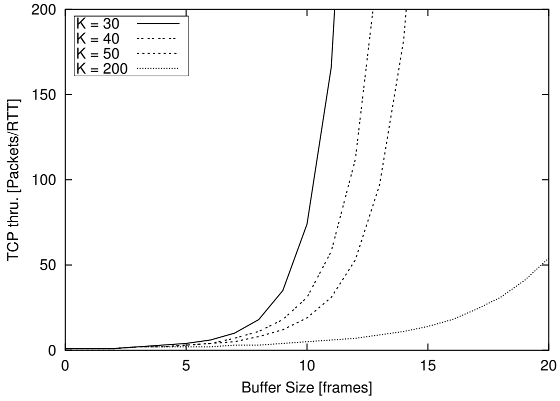

Figure 5. Throughput of a TCP connection through a PRS switch as a function of the buffer capacity at 90% load under unconfirmed reservation signaling.

In case of the single-hop PRS network in Figure 3, suppose the channel load is ρ, a buffer size of q frames, K competing trunks, and one packet per frame, then we have $\ell = \beta(K, \rho, q)$, provided that request blocking for any trunk follows a Bernoulli trials process with blocking (success) probability $\beta(K, \rho, q)$. Hence the average TCP throughput, in units of packets per RTT is given by

$$\text{TCP throughput} = \frac{3}{2\sqrt{\beta(K, \rho, q)}}. \qquad (6)$$

The plot in Figure 5 shows the average TCP throughput across the PRS bottleneck obtained from (6). It indicates that a trunk of TCP connections may not be able to fully utilize its successful reservation requests due to the sensitivity of the throughput to available buffer capacity. This is obvious, for example, at $q = 5$, where the throughput is a few packets per round-trip time (approx. 4 at $K = 30$), despite that more than 90% of the requests are successful according to Figure 4 (≈ 0.3 for a single-hop path). For each value of K, there is a buffer capacity requirement below which the TCP throughput falls rapidly, thus limiting the ability of a trunk to efficiently utilize its bandwidth allocation, assuming a sufficiently large connection round-trip time. This is especially true when the trunk consists of few high-demand flows carrying the bulk of the traffic and a large number of short-lived connections—a characteristic of Internet traffic.

The sensitivity of the TCP throughput to the buffer size is a result TCP's additive-increase, multiplicative-decrease control rule. TCP transmits a number of packets corresponding to the current window size every round-trip time. It interprets packet loss within a transmission window as a congestion signal and reacts by halving the window size to ease-up the congestion. In PRS, blocking (and hence packet loss) depends on the load ρ and not on the actual sending rate of the TCP connections across the bottleneck. Even though there is no congestion, i.e., $\rho < 1$, transient overload for durations less than the cycle size are likely to occur due to trunks contending for the same slots. Consequently, the connection under study will suffer loss at small transmission rates causing its transmission window to collapse. This analysis extends to the case of multi-hop path where ℓ is equal to the path blocking probability. Let p_t denote the path of a trunk t, and let c be the number of TCP flows in t. The average bandwidth utilization efficiency for the trunk is

$$U_t^{\text{unconfirmed}} = \frac{1}{b_t} \cdot \frac{3c}{2RTT\sqrt{PB(p_t)}}. \qquad (7)$$

Note the inverse proportionality to the trunk's bandwidth allocation and the dependence on the number of flows, which indicate that the lack of adequate buffering capacity defeats the efficiency goal of end-to-end TCP congestion control–the ability of a single flow to utilize its allocated capacity.

Although Figure 5 shows that traffic regulation in PRS results in modest buffer capacity requirements for TCP to achieve high bandwidth utilization (less than 15 buffer places at $K = 50$), the technological limitations on optical buffering may render such requirements impractical. In the next section, we tackle the problem of maximizing TCP throughput given buffer constraints.

5.2. Elimination of contention-induced losses using confirmed signaling

In this section we investigate the ability of TCP trunks to efficiently utilize their successful reservations when packets are released from the ingress only to confirmed reservations. Since confirmed signaling eliminates packet loss inside the network, the following analysis is similar to that of TCP performance along a single-bottleneck path [13]. Here the bottleneck is the trunk queue at the ingress router.

For simplicity, we shall limit our discussion to the case where only one trunk originates (and the reverse-path trunk terminates) at each edge router, though it is readily applicable to the general case when trunk-based queuing at the ingress is used as shown in Figure 1. Since packets are released to confirmed reservations, no packet loss occurs at the core switches. We shall assume that the trunk buffer at the ingress is of size B packets, where B is large enough to accommodate the burstiness of TCP connections crossing the link, yet smaller than the sum of their receiver-advertised window sizes. As TCP flows continuously probe for additional bandwidth by increasing their transmission window size, loss occurs at an ingress router whenever $\sum_{i=1}^{c} W_i > \mu T + B$, where W_i is the window size of the i^{th} connection, T is the average round-trip time of the connections excluding the time spent by the packets at the ingress buffer, μ is the rate of confirmed (successful) reservations, and c is the number of TCP connections within the trunk. As each TCP connections increases its window size by one segment size (packet) every round-trip time during which it receives no congestion indication, the condition states that loss will occur when the sum of the window sizes exceeds the maximum number of packets that can be simultaneously buffered or in-transit.

In the worst case, all flows will incur packet loss when the ingress becomes congested and the collective throughput will drop to $(\mu T + B)/2$ before increasing linearly to reach $\mu T + B$ in $(\mu T + B)/2c$ round trips, and repeat the congestion cycle. It follows that the average trunk throughput is $\frac{3}{4}(\mu T{+}B)$ per round-trip time. Typically this form of synchronization among flows is avoided through the use of active queue management schemes such as Random Early Detection (RED) [7] which drops incoming packets with a increasing probability when queue occupancy crosses a certain threshold.

As the time spent by a packet in the ingress queue is at most $\frac{B}{\mu}$, we have $RTT \leq T + B/\mu$. Thus, using confirmed reservation scheme, the average collective throughput is always better than $0.75(\mu T + B)/RTT \geq 0.75\mu$. Once again, let p_t denote the path of the forward trunk. Assuming one packet per frame, a path blocking rate equal to $PB(p_t)$, and no loss of packets on the control plane, the rate of successful reservations is given by $\mu = b_t(1 - PB(p_t))$, where b_t is the trunk's bandwidth allocation in $\mathrm{frames/s}$. Thus, under a confirmed signaling scheme, the trunk utilization efficiency is

$$U_t^{\mathrm{confirmed}} \geq \frac{3}{4}\frac{\mu}{b_t} = \frac{3}{4}(1 - PB(p_t)). \tag{8}$$

Comparing (8) with the bound in (3), we find that confirmed reservation signaling results in near optimal bandwidth utilization efficiency. Note that, in contrast with the unconfirmed signaling case (Equation (7)), efficiency does not depend on the number of flows within the trunks, and is not affected by the size of the bandwidth allocation, nor the round-trip time.

6. Simulation results and discussion

In Section 5, we combined approximate models of a PRS network and TCP congestion control to show that using an unconfirmed reservation scheme, given a sufficiently large round-trip time, TCP trunks will not be able to efficiently utilize their successful reservations under stringent buffer capacity constraints, and that a confirmed reservation scheme ensures good utilization of successful reservations independently of the round-trip time. In this section we support these findings using ns-2 simulations [1], which we augmented with an implementation of PRS. In the first set of experiments, we simulate a single-hop network to verify that TCP performance under the confirmed and unconfirmed schemes throughput performance agree with the analytic results. We also evaluate the effects of reducing the number of high-demand flows within a trunk, increasing the number of packets per frame, and the benefit of confirmed reservation on the performance of web-like transfers. In the second set of experiments we evaluate the blocking rate over paths of increasing lengths and verify the effectiveness of the confirmed-reservation scheme in achieving almost full utilization of successful reservations.

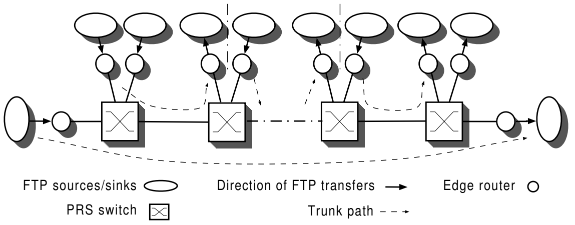

Figure 6. Simulated PRS network.

In earlier sections, the blocking probability (and thus, the rate of successful reservations) was derived assuming buffer capacity constraints; here we use simulation to evaluate the blocking rate under lookahead constraints. Recall from Section 2.3 that a lookahead constraint implies a buffer capacity constraint of the same magnitude, hence, we use the symbol q, used earlier to denote the available buffer capacity, to refer to the value of the maximum permissible lookahead.

6.1. Simulation setup

Figure 6 shows a diagram of the simulated topology. The topology is designed to maximize blocking for the ingress–egress trunks under study. The number of trunks competing at each bottleneck is 30, all of them having identical bandwidth allocations. Each edge router participates as an ingress for a trunk and egress for the trunk on the reverse path. The number of hops of a topology is the number of links counting only those between PRS switches. In a single-hop topology (see Figure 3), all ingress routers are connected to one switch and the egress routers to the other. In this case we are mainly interested in the bandwidth utilization efficiency of the bottleneck link. In the multi-hop configuration, only one trunk crosses all bottlenecks and at each bottleneck it competes with 29 trunks that run for only one hop. In this configuration we measure the bandwidth utilization efficiency of the long-haul trunk.

We ran scaled down simulations to reduce the simulation time without affecting the validity of the results [16]. Link bandwidth is taken to be $20\ \mathrm{Mb/s}$ and each trunk is composed of 20 FTP clients unless otherwise specified. The receiver-advertised window is set to 400 packets so that a single connection can consume the trunk's bandwidth allocation. The default frame size is one FTP packet ($512\ \mathrm{B}$), and the default end-to-end propagation delay of the longest path is $10\ \mathrm{ms}$ with all links having the same propagation delay. Ingress routers have RED queues and their buffer size is set to five times the bandwidth–delay product along the trunk's path.

The blocking rate of a trunk is the complement of the ratio of confirmed reservations to the total reservation requested by the trunk's ingress over the simulated time. The bandwidth utilization efficiency of a trunk is the sum of

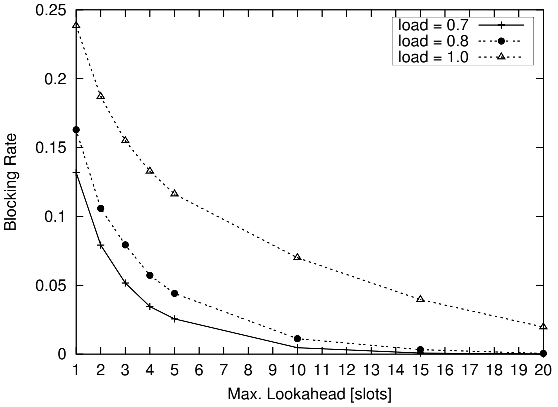

Figure 7. Blocking rate in a single-hop PRS network at different values of ρ.

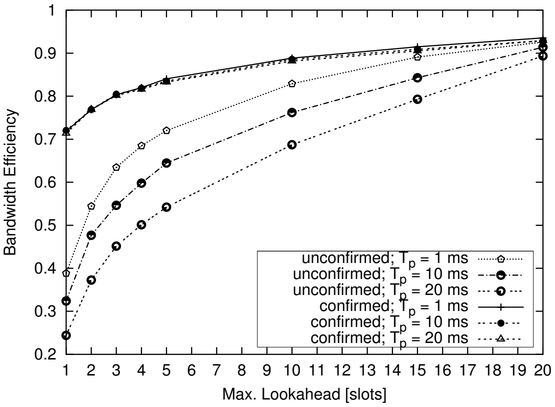

Figure 8. Bandwidth efficiency at 100% load.

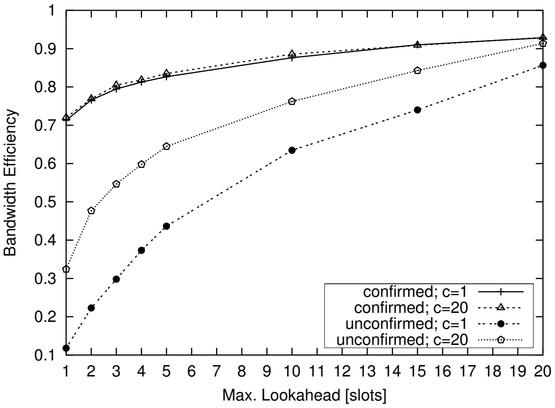

Figure 9. Effect of number of connections within a trunk.

the time-average end-to-end TCP goodput of its component flows divided by its bandwidth allocation (both in b/s). All experiments ran for 30 seconds of simulated time and were repeated until the 95^{th}-percentile confidence limits extend no more than 5% on either sides of the reported point estimate.

6.2. Sensitivity of bandwidth efficiency to buffer constraints

In this set of experiments we verify the sensitivity of bandwidth utilization efficiency to buffer constraints in the case of unconfirmed reservation scheme and the effectiveness of the confirmed scheme in achieving high utilization of the successful reservations independently of the round-trip time. Figure 7 shows the average blocking rate across the bottleneck as a function of the maximum lookahead at different values of link utilization (ρ). Figure 8 compares the collective bandwidth utilization efficiency of the 30 trunks across using the confirmed and unconfirmed reservations schemes at $\rho = 1$ and different value of the one-way propagation delay T_p.[3] Observe the throughput under the latter deteriorates and almost becomes zero given stringent buffer constraints, whereas the TCP connections using the confirmed reservations are always able to fully utilize successful reservations independently of the round-trip delay. For example, at lookahead constraint buffer of 1 frame, the throughput across the bottleneck is slightly less than the ratio of successful reservations $(1 - 0.24) = 0.76$ (Figure 7). Varying the size the PRS frame from 1 to 10 FTP packets did not result in any significant change in the throughput of the TCP connections.

Figure 9 shows the effect of decreasing the number of flows within a trunk, c, on the sensitivity of the unconfirmed and confirmed schemes to the available buffer capacity. Here we set the ingress buffer capacity so that no loss occurs at the ingress hence isolating the effect of bottleneck buffer capacity and measured the collective TCP goodput of the trunks. The confirmed scheme is virtually unaffected by the reduction in number of flows as opposed to the unconfirmed scheme whose bandwidth utilization efficiency drops by more than 30% at 5 buffer places. We conclude that in applications involving a small number of high-bandwidth connections and stringent buffer constraints, the confirmed reservation scheme is necessary to achieve good bandwidth utilization.

6.3. Performance of short-lived flows

Experiments so far have emphasized the ability of long-lived flows to efficiently utilize available bandwidth. In this section, we turn our attention to the performance of short-lived flows typical of web-transfers. We are particularly interested in the flow time of short-lived flows. The flow time is the time is takes the flow to complete its transfer so that all packets are delivered at the FTP receiver. Since packet loss results in reduction in the TCP transmission window size, it increases the flow time of the connection. Intuitively, the elimination of contention-induced loss using the confirmed reservation scheme results in improved flow-time. However, this may be offset by the waiting time

[3] Note that T_p relates to the round-trip delay as $T = 2T_p + T_q$, where T_q is the sum of queuing delays along the forward and the reverse paths.

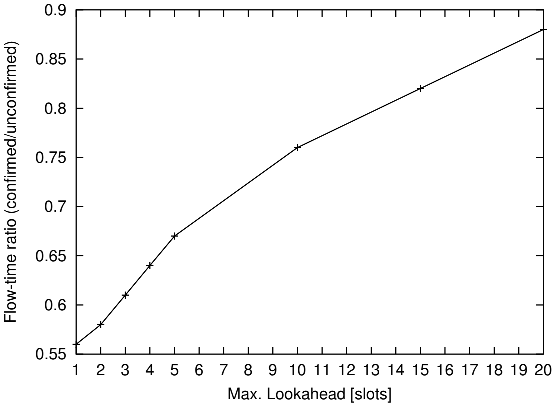

Figure 10. Ratio of average flow time under the confirmed and unconfirmed schemes.

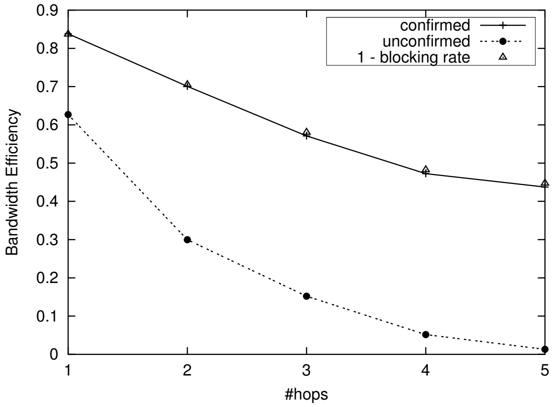

Figure 11. Rate of successful reservations and bandwidth utilization efficiency of the long-haul trunk at $\rho = 1.0$ and $\mathbf{q = 5}$.

at the ingress in case the blocking rate is high.

In the following set of experiments, a trunk is made up entirely of short-lived flows that arrive according to a Poisson process with rate 20 arrivals per second. The size of a short-lived flow is Pareto-distributed with mean of 120 packets and shape parameter set to 1.35 to match values for typical web transfers [3].

Figure 10 shows a plot of the ratio of average flow time under the confirmed and unconfirmed reservation scheme respectively. Observe that the flow time is improved by almost 50% under the confirmed scheme.

6.4. Effect of path length on bandwidth efficiency

The following set of experiments evaluates the effect of the lookahead constraints on the maximum bandwidth utilization efficiency, and the effectiveness of the confirmed scheme over multi-hop paths.

We varied the number of hops along the path of the long-haul trunk from 1 to 5 and studied the effects of buffer constraints on the bandwidth utilization efficiency. Figure 11 and 12, are plots of the bandwidth utilization efficiency and the rate of successful reservations of the trunk against its path length when ρ on all hops is 1.0 and 0.8 respectively and the lookahead is 5 channel slots. We can see that the confirmed scheme achieves almost full utilization of successful reservations. These experiments were designed to isolate the effects of lookahead constraints on blocking performance. The mismatch between the simulation results and the bounds in Figure 4 is either due to the distortion of the pattern of requested time slots by a trunk, which results from buffering at the the core switches, or due to the lookahead constraint. Since the 29 background trunks at each bottleneck run for only one hop, the long-haul trunk is guaranteed to compete with periodic traffic at each hop. Only the request pattern of the long-haul trunk is distorted as it traverses one hop to the next. Observe that the lookahead constraint is set to 5, and so is the number of hops, while the original period of the trunk requests is at least 30 channel slots. This implies that at no point in these experiments, two or more requests from the long-haul trunk compete with one another for buffer or channel resources. Thus, in this set of experiments, even with the absence of shaping at the core switches, the blocking performance of the long-haul trunk is affected only by the lookahead constraints.

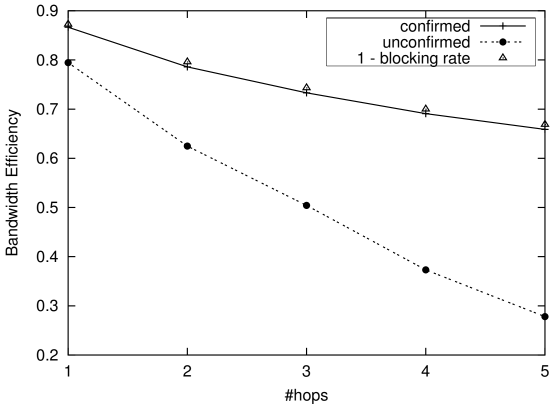

Figure 12. Rate of successful reservations and bandwidth utilization efficiency of long-haul trunk at $\rho = 0.8$ and $\mathbf{q = 5}$.

Comparing the blocking probability at different number of hops for $\rho = 0.8$ and $K = 30$ $(M = 37)$ in Figure 4 with the observed blocking rate in Figure 12, we find that the addition of lookahead constraints almost triples the blocking rate. For instance, when the number of hops is 5, the observed blocking rate is ≈ 0.32, while the blocking probability from Figure 4 is close to 0.11. We conclude that the lookahead constraints need to be taken into account at the provisioning and traffic engineering phases.

7. Concluding remarks

PRS is a reservation-based switching architecture that performs transmission control by regulating traffic at the ingress edges and at the core to minimize the buffer ca-

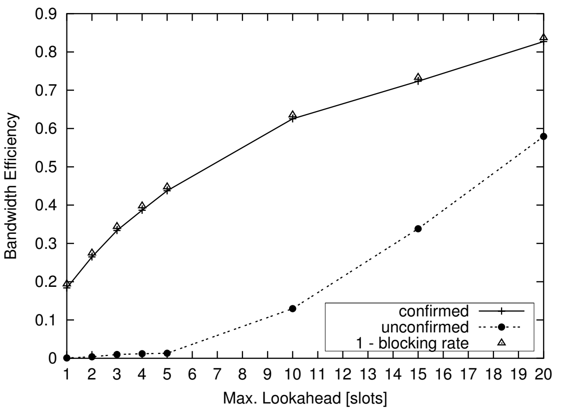

Figure 13. Rate of successful reservations and bandwidth utilization efficiency of long-haul trunk at $\rho = 1.0$ and path length of 5 hops.

pacity required for TCP traffic to achieve high bandwidth utilization of network links. However, due to technological limitation, these buffer requirement may be too expensive or impractical. In this paper we have considered the dual problem of maximizing bandwidth utilization efficiency given buffer capacity constraints.

First, we established an upper-bound on the maximum bandwidth utilization efficiency that is achievable by a trunk routed along a path of a given number of hops. This bound can be used at the provisioning and traffic engineering phases to trade off the number of trunks competing at the channels to the length of the paths crossing the channel, given a desired level of bandwidth efficiency and buffer capacity constraints.

Next, we showed that TCP trunks may not be able to fully utilize their successful reservation due to contention-induced packet loss. We proposed a confirmed reservation signaling scheme and evaluated its performance using TCP-driven simulation. Simulation results support the analytical results by showing that the confirmed scheme is effective at maximizing the bandwidth utilization efficiency.

Confirmed signaling maximizes the bandwidth utilization efficiency in the sense that further improvement must be achieved through the reduction of blocking probability along the network's paths.

References

[1] The network simulator ns-2. `http://www.isi.edu/nsnam/ns/`.

[2] M. Baldi and Y. Ofek. Fractional lambda switching. In *IEEE International Conference on Communications (ICC2002), Optical Networking Symposium*, pages 2692–2696, April 2002.

[3] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, 1997.

[4] P. H. Dana. Global positioning system (GPS) time dissemination for real-time applications. *Real-time Systems*, (12):9–40, 1997.

[5] M. Elhaddad, R. Melhem, T. Znati, and D. Basak. Traffic shaping and scheduling for OBS-based backbones. In *OPTICOMM 2003, Proceedings of SPIE*, volume 5285, pages 357–368.

[6] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. Internet RFC 2582, IETF, April 1999.

[7] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, 1993.

[8] S. Herrería-Alonso, A. Suárez-González, M. Fernández-Veiga, R. Rodríguez-Rubio, and C. López-García. Improving aggregate flow control in differentiated services networks. *Computer Networks*, 44(4):499–512, March 2004.

[9] 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):81–95, 1993.

[10] D. K. Hunter, M. C. Chia, and I. Andonovic. Buffering in optical packet switches. *IEEE Journal of Lightwave Technology*, 16(12):2081–2094, December 1998.

[11] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwith-delay product networks. In *ACM SIGCOMM*, Pittsburgh, Pennsylvania, 2002.

[12] H. T. Kung and S. Wang. TCP trunking: Design, implementation, and performance. In *IEEE ICNP*, Toronto, Canada, 1999.

[13] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. on Networking*, 5(3):336–350, June 1997.

[14] A. E. Ozdaglar and D. P. Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Trans. Netw.*, 11(2):259–272, 2003.

[15] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Trans. Netw.*, 8(2):133–145, 2000.

[16] K. Psounis, R. Pan, B. Prabhakar, and D. Wischik. The scaling hypothesis: Simplifying the prediction of network performance using scaled-down simulations. *Computer Communication Review*, 33(1):35–40, Jan. 2003.

[17] J. Ramamirtham and J. S. Turner. Time sliced optical burst switching. In *IEEE INFOCOM 2003*.

[18] J. W. Roberts and J. T. Virtamo. The superposition of periodic cell arrival streams in an ATM multiplexer. *IEEE Trans. Commun.*, 39(2):298–303, Feb. 1991.

[19] K. Thompson, G. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, 1997.

[20] J. S. Turner. Terabit burst switching. *Int'l J. High-Speed Networks*, 8(1):3–16, 1999.

[21] J. Y. Wei and R. I. McFarland. Just-In-Time signaling for WDM optcial burst switching networks. *IEEE journal of lightwave technology*, 18(12):2019–2037, December 2000.

[22] Y. Xiong, M. Vandenhoute, and H. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE journal on selected areas in communications*, 18(10):1838–1851, October 2000.

[23] 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*, pages 79–90, 1997.

[24] 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.