

# Analysis of a Transmission Scheduling Algorithm for Supporting Bandwidth Guarantees in Bufferless Networks

Mahmoud Elhaddad, Rami Melhem, and Taieb Znati<sup>1</sup>

## Abstract

In a network of bufferless packet multiplexers, the user-perceived capacity of an ingress–egress tunnel (connection) may degrade quickly with increasing path length. This is due to the compounding of transmission blocking probabilities along the path of the connection, even when the links are not overloaded. In such an environment, providing users (e.g., client ISPs) with tunnels of statistically guaranteed bandwidth may limit the network’s connection-carrying capacity.

In this paper, we introduce and analyze a transmission-scheduling algorithm that employs randomization and traffic regulation at the ingress, and batch scheduling at the links. The algorithm ensures that a fraction of transmissions from each connection is consistently subject to small blocking probability at every link, so that these transmissions are likely to survive long paths. For this algorithm, we obtain tight bounds on the expectation and tail probability of the blocking rate of any ingress–egress connection. We compare the bounds to those obtained using the FCFS link-scheduling rule. We find that the proposed scheduling algorithm significantly improves the network’s connection-carrying capacity.

In deriving the desired bounds, we develop an analytic framework for stochastically comparing network-wide routing and bandwidth allocation scenarios with respect to blocking in a packet multiplexer. The framework enables us to formally characterize the routing and bandwidth allocation scenarios that maximize the expected blocking rate along the path of a tagged connection.

## 1 Introduction

Transit networks often provide a “virtual leased-line” service in the form of ingress–egress tunnels (connections) of guaranteed capacity. The user of the transit network, for example, a lower-tier Internet Service Provider, is guaranteed a statistically bounded and small transmission blocking rate (e.g., fraction of lost packets) as long as it injects traffic into the tunnel without violating a certain input traffic characterization.

Given a desired statistical (upper) bound on the blocking rate of all connections, the *connection-carrying capacity* of

a network is the number of connections that the network can serve (hence network utilization) without violating the desired bound. The virtual leased-line model offers significant improvements in the connection-carrying capacity of transit networks compared to services based on lossless multiplexing (e.g., TDM-based leased lines) [15].

The connection-carrying capacity of a network depends on the amount of buffering available for contention resolution at the core nodes (routers or switches). This is true for every work-conserving link scheduling rule (e.g., FCFS). However, some scheduling rules may perform better than others with limited buffers. In other words, for a given router buffering capacity, a network’s connection-carrying capacity depends on the link-scheduling algorithm used at the core routers.

In this paper, we consider the problem of transmission scheduling in networks of time-slotted bufferless links to provide virtual leased-line services—efficiently in terms of connection-carrying capacity. We assume the connections are permanent: once admitted a connection remains in the network. Thus the scheduling is one of maximizing the number of connections the network can admit without violating the blocking rate guarantees. The links are bufferless in the sense that buffering can be used for alignment of arriving packets with time-slot boundaries but not for contention resolution. The focus on networks with small or no contention-resolution buffers is motivated by the emergence of high-speed routers using on-chip static RAM implementation of packet buffers [1], and all-optical packet switches where buffers can be implemented using bulky fiber delay lines [12]. We limit this exposition to the bufferless case, as opposed to the case where links may have very small buffers, in order to sidestep the analytical complications caused by the loss of traffic characterization upon traversing a buffered link [9, 14].

For the class of networks considered in this paper, links are modeled as follows: define the *speedup* of a link as the maximum number of constant-size packets that can be simultaneously served by the link during a time slot. Each link is equipped with a number of *alignment* buffers equal to the link’s speedup. Packets arriving at the head of a link within a time slot are stored in the alignment buffers. The stored packets are then transmitted at the beginning of the following link slot. This model is representative of optical fiber links in a transit network. A link with speedup  $s$  models an optical fiber link that carries  $s$  wavelength channels of identical capacity

<sup>1</sup>Department of Computer Science, University of Pittsburgh. Pittsburgh, PA 15260. USA. Email: {elhaddad,melhem,znati}@cs.pitt.edu.

through wavelength division multiplexing (WDM). Improved speedup is not without cost: incrementing the speedup of a link requires an additional optical transceiver at each of the link’s ends.

For bufferless links, FCFS scheduling degenerates into a natural scheduling algorithm that we refer to as the Earliest Arrivals rule (EA). In EA, packets arriving within a slot are considered for transmission in their order of arrival. If the number of packets arriving at the link within a time slot exceeds the speedup, excess packets are blocked. Careful reading of the results in [14] shows that EA would result in good connection-carrying capacity only when links have large speedup values. At small values of speedup, the transmission-blocking rate of a connection deteriorates quickly with increasing path length, even when links are lightly loaded.

The high cost of optical transceivers needed to increase the speedup of a link raises the question of whether there are scheduling algorithms that support good connection-carrying capacity at small speedup values. In this paper, we present such an algorithm in the context of a reservation-based transmission control framework that is of practical importance in networks carrying loss-sensitive (e.g., TCP) traffic. Before presenting an overview of the proposed algorithm and the analytical results, we describe the reservation-based framework and state how the EA scheduling rule applies in its context.

The reservation-based framework implements virtual leased lines as follows: each connection regularly generates transmission requests for future time slots along its ingress–egress path. A request may be blocked due to contention at any link. If a request is not blocked, the egress sends an acknowledgment back to the ingress. Packets entering the transit network are buffered at the ingress and grouped into slot-sized data frames. The data frames are then released into the bufferless network according to the schedule of acknowledged reservations. In the context of this framework, the order of arrivals at a link in the EA rule is the order of receiving the transmission requests for a particular time slot (as opposed to the order of arrival of the corresponding data frames).

Observe that with acknowledgments each ingress–egress connection behaves as a lossless pipe of capacity equal to the rate of successfully acknowledged transmission requests. Loss can only occur at the ingress due to buffer overflow. If the ingress is equipped with appropriately large buffer, loss occurs only due to the difference between the packet arrival rate and the rate of successful reservations over a time scale much longer than a time slot duration. This allows loss-sensitive flows such as TCP and its high-speed variants [5, 7] to fully utilize the virtual pipe of successful reservations. Given an expression for the tail of the blocking rate distribution of a connection, the network operator can overprovision the bandwidth allocated to the connection so that the expected rate of successful reservations is arbitrarily close to the nominal connection bandwidth allocation. Additional details about the reservation-based framework can be found in [4].

## 1.1 Overview of Proposed Solution

As an alternative to EA, we introduce a scheduling algorithm, BATCH, where, at every link, transmission requests from each connection are presented to the link in batches. Each batch covers a future *reservation epoch* of fixed duration. Transmission requests are scheduled at a link in the order of reception of the corresponding batches.

BATCH specifies rules for generating time slot requests at the ingress and for scheduling requests at core links. It also requires the ingress to perform “phase randomization” to reduce the likelihood of synchronized (completely overlapping) reservation epochs among different connections. BATCH improves the transmission-blocking rate for connections traversing a large number of hops by guaranteeing that a subset of transmissions from each connection are consistently subject to small blocking probability at every hop—even when the expected blocking rate at the links is comparatively high.

## 1.2 Results and Contributions

The paper reports quantitative and qualitative results characterizing the performance of BATCH. We obtain a tight bound on the expected blocking rate of any network connection under both EA and BATCH as a function of the speedup and load at the links. Moreover, for BATCH, we obtain a bound on the tail probability of the blocking rate.

The expected blocking rate bound defines a tradeoff between the path length and link utilization. The tradeoff is of practical use in deriving path length and load constraints on connection routing. Numerical results show that, at small values of link speedup, the algorithm achieves significantly better tradeoffs (hence connection-carrying capacity) compared to EA.

The qualitative results are summarized as follows. First, we show that BATCH ensures correlation among scheduling decisions at different links so that a subset of the transmission requests in each reservation epoch are subject to blocking probability less than the expected blocking rate at every hop (Theorem 2, Section 8). Second, BATCH is fair in the sense that connections sharing a link have equal bounds on the expected blocking rate at that link (Theorem 3, Section 8). The fairness result implies that BATCH does not give favorable treatment to connection traversing long paths at the cost of penalizing other connections.

Given the load at the links, the bounding analyses require the characterization of “worst-case” routing and bandwidth allocations scenarios (regimes) that maximize the expected blocking rate for a tagged connection at every link it traverses.<sup>1</sup> Toward this end, we define a stochastic order relation among the regimes that result in a prescribed load value at a particular link. The order relation is defined with respect to the distribution of *contention* at any time slot (number of requests for a link time slot). This analytic framework leads to necessary

<sup>1</sup>Because of the fairness property, the expected blocking rate over all connections sharing a link is the same as the expected blocking rate of the tagged connection at that link.

and sufficient conditions for maximal expected request blocking rate (i.e., maximum given the value of the load) at the link. This characterization of worst-case regimes provides a proof of the observation in [16] that the buffer overflow probability in a multiplexer of regulated streams is largest whenever each stream has the minimum bandwidth allocation.

The analysis yields a product-form bound for the blocking probability of individual transmission requests in terms of the blocking probability at the link. Analysis of the request blocking probabilities at a link builds upon the blocking probability in a constant-service time multiplexer of periodic streams. The blocking rate bounds are tight since they represent the performance of a connection routed along a set of links where the conditions for maximal expected blocking rate are satisfied.

### 1.3 Related work

Research related to resource management in networks with limited buffers within the optical packet and burst switching communities focuses primarily on system performance metrics, such as switch and network throughput as opposed to quality of service guarantees (See [2, 12] and references therein). When links are slotted and without contention resolution buffers, Optical Burst Switching (OBS) is a reservation-based transmission control scheme that implements the EA scheduling rule.

Research on statistical quality of service guarantees focuses mostly on devising scheduling strategies that support packet delay guarantees assuming that contention resolution buffers are large enough to prevent packet loss (for example, see [10] and reference therein).

Work on statistical loss guarantees typically aims at deriving bounds on the tail probability of the queue size at a link, given an input traffic characterization. The bounds are then used in dimensioning the link buffer size to achieve a desired loss rate [8, 11, 18]. Today, transit networks providing virtual leased-line services rely on this buffer-provisioning approach. Obviously, it is not applicable when buffer capacity is limited by technological constraints.

The work by Reisslein et al. [14] provides a bufferless-multiplexing framework for supporting statistical delay guarantees in multihop networks. Using traffic regulation at the ingress and bufferless multiplexing at the core, they transform the problem of providing ingress–egress delay guarantees into one of providing loss guarantees. The loss bounds are obtained using an approximate fluid-multiplexer model. In the next section, we argue that the fluid model may severely underestimate the blocking probability in packet multiplexers, and specify the speedup-based model adopted in this paper.

The remainder of the paper is organized as follows. We characterize the performance of the EA rule under the speedup-based link model in Section 3. In Section 4, we present the

proposed transmission-scheduling algorithm, BATCH. In Section 5, we identify conditions on routing and bandwidth allocation that result in maximal expected blocking rate at every link along the path of a tagged connection. Bounds on the request blocking probabilities at a link are derived in Section 6, and bounds on the expectation of the blocking rate and its tail probability are obtained in Section 7. The correctness and fairness results are presented in Section 8, followed by numerical results in Section 9. Finally, we present our concluding remarks in Section 10.

## 2 Bufferless Multiplexing Model

Blocking in a network of bufferless multiplexers (links) has been analyzed in [14] using an approximate fluid-multiplexing model. The fluid model builds on the fundamental assumption that, at each (arbitrarily small) time step, a link can simultaneously serve traffic from any number of connections as long as the sum of their instantaneous transmission rates does not exceed the link capacity. In this section, we identify some limitations of the fluid model and introduce the link model used in this paper.

### Limitations of the fluid model

When used to predict the performance of bufferless multiplexers of packetized traffic, the fluid model may underestimate the blocking probability: If the multiplexer serves packets sequentially (as a single server system), the fluid assumption (stated above) does not hold for time steps of length equal to or smaller than the packet transmission time. As a result, the fluid model fails to account for blocking due to multiple arrivals within a packet transmission time.

The quality of the fluid approximation is sensitive to the relation of the link capacity to connection peak transmission rates. This follows from the fact the peak transmission rate of a connection can be chosen arbitrarily and remains constant across all time scales. For example, if the link capacity is larger than the sum of peak transmission rates for all connections sharing the link, blocking never occurs in the fluid service model irrespective of the load.<sup>2</sup> This is again in contrast to the behavior of packet multiplexers where the peak transmission rate of every connection is—by force—equal to the link capacity at time scales shorter than or equal to the packet transmission time (time slot). Under heavy load, blocking occurs frequently in a bufferless packet multiplexer due to the coincidental arrival of multiple packets within a time slot.

Due to the above limitations, the fluid approximation is inadequate for computing bounds on the blocking performance in networks of bufferless packet multiplexers. However, the performance bounds obtained using the fluid model in [14] can be interpreted as highlighting the performance gains that can be obtained by provisioning each link to act as a multi-

<sup>2</sup>In this case, the load can be made arbitrarily close to 1 without incurring any traffic loss by having connections transmit at peak rate more frequently.

server system capable of serving multiple packets in each time slot. The issue becomes whether the scheduling rule affect the amount of overprovisioning needed to achieve the desired performance bounds.

### Bufferless packet multiplexers with speedup

Now, we extend the statistical benefits of simultaneously serving multiple connections to packet multiplexers by explicitly modeling the link as a time-slotted multi-server system. This link model enables us to differentiate among link scheduling rules with respect to the number servers (e.g., optical transceivers) required per link to achieve a desired blocking performance.

During each time slot, the instantaneous rate of a connection is a binary random variable assuming the value of 1 if the connection requests transmission during that slot, and 0 otherwise. A link can serve transmissions from a number of connections not exceeding its *speedup* value. Under this speedup model, a bufferless link is a multi-server system where the number of servers is equal to the speedup, and all servers have the same constant service time.

We demonstrate the role of speedup using an example that later turns out being of particular interest. Consider a link multiplexing  $N$  periodic connections. Each connection offers to the link one packet exactly every  $M$  slots. Taking any  $M$  consecutive link slots (a period of the link) as a reference, each connection is equally likely to offer its packet during any of the  $M$  slots. Furthermore, the slot choice of each connection is independent from other connections. Let  $s$  be speedup of the link. Then, for a given load  $\nu \in \{0, 1/M, 2/M, \dots, (M-1)/M\}$ , the number of connections,  $N$ , has to satisfy  $N = \nu sM$ .

Suppose that speedup is increased by a factor  $f$  in order to scale the link capacity by the same factor. The load can be kept the same by either (1) increasing the number of connections  $N$  by a factor  $f$  while keeping  $M$  fixed, or (2) keeping the number of connections fixed by increasing their bandwidth allocations by a factor  $f$ , which is equivalent to shrinking  $M$  by the same factor (given that in the current example, each connection requests exactly 1 slot every  $M$ ). In either case, increasing the speedup at a fixed load results in improved link blocking performance. The improvements in the first case can be established only in the limit as  $s$  (and by proportionality  $N$ ) approach  $\infty$ , for example using CLT. Since we are contemplating the problem of multiplexing ingress–egress traffic aggregates, which are naturally limited in number, we are not interested in such limiting regimes.

For the second case, suppose the slots within the reference period are numbered 1 through  $M$ . Let  $\gamma_{i,j}$  be a bernoulli random variable that assumes the value 1 if connection  $i$  chooses slot  $j$  (i.e., with probability  $1/M$ ). Then the number of connections requesting slot  $j$  is  $\gamma_j \triangleq \sum_{i=1}^N \gamma_{i,j}$ . By the Chernoff bound, for  $1/2 \leq \nu < 1$ :  $\Pr\{\gamma_j > s\} < e^{-s(1-\nu)^2/(3\nu)}$  which drops quickly as  $s$  increases.

As mentioned previously, the high cost of optical transceivers necessary to activate additional wavelengths (that is increase the speedup of a link) drive the interest in scheduling rules that perform well at relatively small values of speedup. In the next section we find that the Earliest Arrivals (EA) rule does not satisfy this requirement.

### 3 Performance of the EA Rule

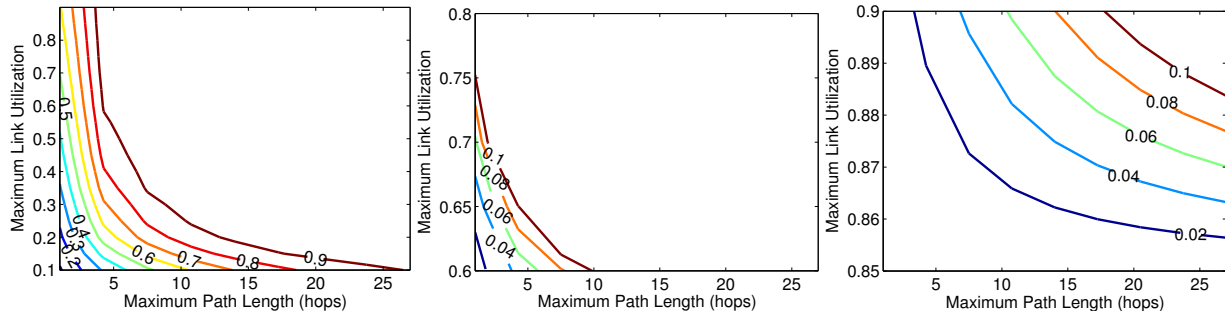
Consider the example introduced in the previous section. Under the EA rule, the probability of blocking a packet (or transmission request) from a tagged connection is bounded by the probability that  $s$  or more connections, out of the remaining  $N-1$  connections, choose the same slot as the tagged connection.<sup>3</sup> Denoting this probability with  $\beta$ , we have:

$$\beta = \sum_{i=s}^{N-1} \binom{N-1}{i} \left(\frac{1}{M}\right)^i \left(1 - \frac{1}{M}\right)^{N-1-i}. \quad (1)$$

Suppose that packets from a given connection are blocked at each link according to an independent bernoulli trial. Then the probability of blocking of a packet along a path of  $h$  links (identical to the one under consideration) is bounded by  $1 - (1 - \beta)^h$ . This is also the expected value of the blocking rate (fraction of blocked packets) experienced by a connection. In Section 5, we find that the example above represents a worst-case analysis of the blocking performance under EA when traffic entering the network is shaped at the ingress into periodic streams to minimize the blocking rate.

To visualize the performance of EA in terms of a network’s connection-carrying capacity, we plot the tradeoff between the link utilization and the path length to maintain a desired bound on the expected blocking rate. We refer to such tradeoffs as “ $U$ - $P$  tradeoffs.” Poor tradeoffs indicate that the network can provide blocking guarantees only at light link loads (small number of connections sharing each link) and prevent routing traffic over long paths, thus implying poor network utilization. Figure 1 shows the  $U$ - $P$  tradeoffs at different value of the link speedup  $s$ . They are obtained by fixing  $N = 128$  connections and setting the period  $M$  according to the speedup and the desired link load. The plots indicate that speedup plays a major role in determining the network’s connection-carrying capacity. For instance, at  $s = 1$  and link load of only 10%, the network cannot support an expected blocking rate of 0.1 for connections traversing more than a few hops. On the other hand, when  $s = 100$ , the network can support an expected blocking rate bound of 0.02 at link utilization above 85% for connections traversing more than 25 hops. The middle plot shows that, even at speedup as large as 20, supporting an expected blocking rate bound of 0.1 severely limits the diameter of the network.

<sup>3</sup>Since EA is defined for traditional and reservation-based packet switches, we use the term *packet* to refer to actual packets or to the corresponding transmission requests.



**Figure 1:** Link utilization versus path length ( $U$ - $P$ ) tradeoffs for link capacity 128 slots/period and different values for link speedup  $s$ . Left: link speedup  $s = 1$ , middle:  $s = 20$ , right:  $s = 100$ . Each contour is the locus of a constant expected blocking rate in the  $U$ - $P$  plane.

In the remainder of the paper, we introduce an alternative transmission-scheduling algorithm and compare the  $U$ - $P$  tradeoffs it supports with those of EA. We find that the proposed algorithm supports significantly better tradeoffs.

#### Remarks.

The bound in (1) can be improved by assuming that the time-slot request by the tagged connection is equally likely to be received in any order among the  $i + 1$  connections contending for the slot (i.e., by multiplying the summand by  $\frac{i+1-s}{i+1}$ ). However, this assumption does not hold in general due to traffic phase effects that arise in multiplexing regulated traffic (See [6] and references therein).

Though the EA scheduling rule can be modified to realize the assumption by randomly accepting a subset of length  $s$  from the  $i + 1$  transmission requests contending for the slot, this may lead to practical complications. In the reservation-based framework described at the beginning of this paper, if a request arrives for a time slot that already has  $s$  scheduled packets, a victim would have to be chosen at random from the  $s + 1$  contending requests. Canceling an existing reservation results in wasted resources on downstream links and complicates processing of acknowledgments. A different complication arises in traditional packet switches (without reservations) because alignment buffers are typically implemented at the routers' input interfaces rather than at the output links: choosing a random subset of packets destined to the same output is best done through a central arbiter, which can itself become a bottleneck in routers with large number of interfaces and link speedup.

## 4 The BATCH Scheduling Algorithm

The BATCH scheduling algorithm is defined in the context of the reservation-based transmission framework described earlier. In the reservation framework, the ingress border router of a connection regularly generates requests for time slots at a rate equal to the connection's bandwidth allocation, and releases constant-size data frames into the network according to

the schedule of acknowledged transmission requests.<sup>4</sup>

The transmission requests are transmitted from the ingress router in the form of reservation packets that are processed by core link schedulers as they traverse the connection's path. A link scheduler may block requests due to contention. To avoid contention between data and reservation traffic, we assume, as in burst-switched networks [13, 20], that reservation traffic is transmitted out-of-band over special control channels, and are processed in each router in a dedicated control plane.<sup>5</sup>

BATCH is composed of an ingress algorithm for the generation of transmission requests, and a link scheduling rule.

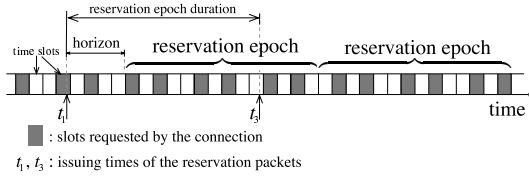
### 4.1 The ingress algorithm

The ingress router periodically sends a reservation packet along the path of the connection. Each reservation packet contains the batch of time slot requests covering a future *reservation epoch* of fixed duration. The time between successive transmissions of reservation packets at the ingress is equal to the duration of a reservation epoch so that a new epoch starts immediately at the end of the preceding one. The duration of time between the transmission of the reservation packet and the start of the corresponding reservation epoch is called the *horizon*. Figure 2 specifies the timing relationships between events in the reservation process for two consecutive reservation epochs at the ingress of a connection. In addition, Figure 3 shows the timing relationships between events for a single epoch over two consecutive links.

The ingress algorithm specifies the requested slots by the connection within each epoch, and performs *phase randomization* as described next.

<sup>4</sup>Arriving packets are grouped into constant-size frames at the ingress router.

<sup>5</sup>The volume of reservation traffic generated by a connection should be made much smaller than the volume of its data traffic by the appropriate choice of the frame size [19]. We assume that the capacity of the control channels and router control planes are dimensioned so that no reservation traffic is lost.



**Figure 2:** Timing diagram of the reservation process at the ingress of a connection for two consecutive epochs.

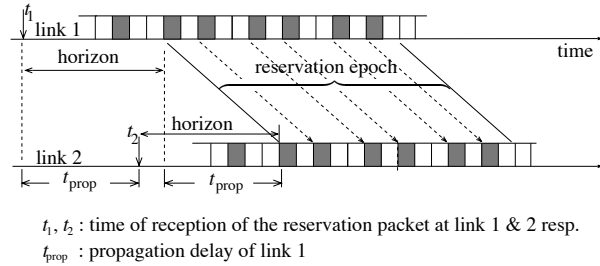
**4.1.1 Pattern of requested slots:** Let  $s\mu$  be the capacity of the first link along the path of a connection  $c$  in frames/second, where  $s$  is the link’s speedup. Since a connection can inject at most one data frame per time slot into the network, from  $c$ ’s point of view the link capacity is simply  $\mu$  slots/second. Further, let  $\rho$  denote  $c$ ’s bandwidth allocation in frames/second. The ingress of connection  $c$  chooses the slots to request according to the staircase function  $\alpha_{\rho,\mu}(t) = \lceil t \frac{\rho}{\mu} \rceil$ ,  $t \geq 0$ , which specifies the number of requested slots in an interval of length  $t$  slots. Precisely, if connection  $c$  requests its first slot at  $\tau_0$ , then for all  $\tau \geq \tau_0$ , connection  $c$  requests slot  $\tau$  iff  $\alpha_{\rho,\mu}(\tau - \tau_0 + 1) > \alpha_{\rho,\mu}(\tau - \tau_0)$ . The staircase pattern limits the variation in the number of slots between two consecutively requested slots to  $\pm 1$ . In the absence of blocking, this minimizes both the maximum and expected inter-transmission delay.

**4.1.2 Phase Randomization:** Upon the initialization of a connection, the ingress picks a phase shift uniformly at random from the length of a reservation epoch. The phase shift is added to the horizon and the time of initialization to obtain the starting slot of the first reservation epoch. Phase randomization ensures that, at every link, reservation epochs from different connections overlap at random phases. It plays an important role in determining the blocking performance of the scheduling algorithm (Section 6).

## 4.2 Link scheduling rule

Core link schedulers sequentially process reservation packets (request batches) in the order of reception. Requests within a batch are also scheduled sequentially or in parallel. Each link scheduler maintains a time-slot availability vector indicating the number of frames scheduled for transmission within each slot in a bounded interval. The number of frames scheduled for transmission during a slot is at most equal to the link speedup,  $s$ . If a new request arrives for a slot having less than  $s$  scheduled frames, the request is granted. Otherwise it is blocked.

The scheduling algorithm is parameterized by a globally defined minimum bandwidth allocation in frames/second and the minimum batch size (number of requests within a batch for a connection having minimum bandwidth allocation). To-



**Figure 3:** Timing diagram showing the pattern of requested slots by a connection on two tandem links.

gether, these parameters determine the duration of a reservation epoch, which is common for all connections. The actual batch size for a connection is determined by the connection’s bandwidth allocation.

It should be clear that the time complexity of the algorithm is  $O(1)$  per transmission request. We discuss the storage complexity of the algorithm in Section 10.

## 5 Analysis Framework

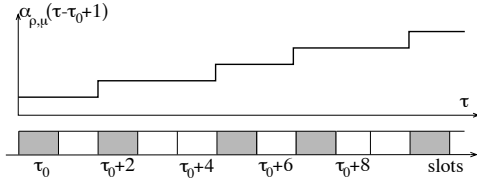
In the remainder of the paper, we analyze BATCH to obtain bounds on the expectation of the blocking rate experienced by an arbitrarily tagged connection and its tail probability. The analysis of blocking is probabilistic due to phase randomization at the ingress routers (Section 4.1.2).

We seek bounds parameterized by the load at each link along the path of the connection. These bounds must be obtained by analyzing the transmission-scheduling algorithm under conditions that maximize the blocking rate at each link, given its load. In this section, we identify necessary and sufficient conditions for maximal blocking rate at a link. We then characterize the network-wide bandwidth allocation and routing scenarios, called *regimes*, under which these condition holds at every link along the connection’s path. The characterization is defined over all possible sets of ingress–egress connections and bandwidth allocations. It is independent of the link-scheduling rule, thus can be used in the analysis of EA as well as BATCH.

We begin by stating our modeling assumptions and establishing some properties of the pattern of slots requested by a connection at any link.

### 5.1 Notation and model assumptions

Let  $\mathcal{L} = \{1, 2, \dots, L\}$  represent the set of links in the network, and let  $\mathcal{C}_l$  be the set of connections sharing link  $l \in \mathcal{L}$ . Let  $\mu_l$  denote the capacity of link  $l$  in slots/second and let  $s_l \geq 1$  be its speedup value. Further, denote the bandwidth allocation of a connection  $c$  by  $\rho_c$  frames/second. A connection can request at most 1 transmission per time slot at any link irrespective of the speedup. Thus we have  $\rho_c \leq \mu_l$  for



**Figure 4:** An example envelope pattern generated by  $\alpha_{\rho_c, \mu_l}(\tau - \tau_0 + 1)$  where  $\rho_c/\mu_l = 2/5$ . Any sequence of consecutive link slots of length 5 is a period of the pattern.

every link  $l$  and  $c \in \mathcal{C}_l$ . The analysis is carried-out under the following assumptions:

- (M1) Links are not oversubscribed:  $\sum_{c \in \mathcal{C}_l} \rho_c < s_l \mu_l$  for all  $l$ .
- (M2) Connection bandwidth allocations and the link capacities are integral multiples of a minimum allocation  $\rho_0$ .
- (M3) Connection phases are statistically independent.

Despite (M1), blocking still occurs whenever the number of connections requesting transmission during a link slot exceeds the link speedup. Since the minimum allocation  $\rho_0$  can be made arbitrarily small, assumption (M2) simplifies the analysis without any loss of generality. Assumption (M3) relates to the randomization of connection phases at the ingress routers. It implies that slots requested by two different connections at a link are statistically independent, unless these connections share an upstream link.

## 5.2 Transmission request patterns in BATCH

We say that the request pattern of a connection  $c$  at some link  $l$  is defined by  $\alpha_{\rho_c, \mu_l}$  if for some starting slot  $\tau_0$  on  $l$ , the pattern matches the one generated from  $\alpha_{\rho_c, \mu_l}$  using the ingress rule (Section 4.1.1).

Let  $(l_1, l_2, \dots, l_h)$  represent the path of a connection  $c$ . The pattern of connection  $c$ 's request on  $l_1$  is defined by  $\alpha_{\rho_c, \mu_{l_1}}$ . It is straightforward to show that this request pattern is periodic.

**Property 1** Let  $T_{c,l}$  be the denominator of the rational number  $\rho_c/\mu_l$  in its simplest form. Then the request pattern defined by  $\alpha_{\rho_c, \mu_l}$  is periodic with period size  $T_{c,l}$  slots.<sup>6,7</sup>

Figure 4 is an example illustrating the periodicity of the envelope pattern when  $\rho_c/\mu_l = 2/5$ .

It is easy to see that the periodicity of the pattern of slots requested by the ingress is maintained across links of different capacity, except for missing requests due to blocking at upstream links.

<sup>6</sup>A function  $f$  is periodic with period  $L$  if for all  $x$ :  $f(x + L) = f(x)$ .

<sup>7</sup>A rational number is in its simplest form when the nominator and the denominator have no common divisors.

**Property 2** Consider two tandem links,  $l_i$  and  $l_{i+1}$ , along the path of a connection  $c$ . If the pattern of slots requested by  $c$  on  $l_i$  is defined by  $\alpha_{\rho_c, \mu_{l_i}}$ , then in the absence of blocking, the pattern of slots requested by  $c$  on  $l_{i+1}$  is defined by  $\alpha_{\rho_c, \mu_{l_{i+1}}}$ .

If the connection experiences blocking upstream of a link  $l$ , the function  $\alpha_{\rho_c, \mu_l}$  defines an envelope pattern for  $c$  at  $l$ : requested slots follow exactly the periodic pattern defined by  $\alpha_{\rho_c, \mu_l}$  except for the requests missing due to blocking upstream.

The following is a direct consequence of Property 1.

**Property 3** The pattern of the requested slots on a link  $l$  due to the superposition of the patterns defined by  $\{\alpha_{\rho_c, \mu_l} | c \in \mathcal{C}_l\}$  is periodic with a period size  $T_l$  equal to the least common multiple of the sizes of the connection periods.

Let  $T_{0,l}$  be the denominator of the rational number  $\frac{\rho_0}{\mu_l}$  in its simplest form. (i.e., the period size of a connection of minimum allocation at link  $l$ ). Then by assumption (M2),  $T_{0,l}$  is an integral multiple of  $T_l$ . We refer to any sequence of length  $T_{0,l}$  of consecutive slots on  $l$  as the period of link  $l$ . We state this as a definition for ease of reference:

**Definition 1 (Link period)** Let  $T_{0,l}$  be the size of the period of a connection of minimum allocation  $\rho_0$  at link  $l$ . Then any sequence of consecutive slots of length  $T_{0,l}$  on  $l$  is called the period of link  $l$ .

Using the properties developed in this section, we now characterize the routing and bandwidth allocation scenarios that lead to maximal blocking rate at a link.

## 5.3 Conditions for maximal expected link blocking rate

As described in Section 2, in our packet-multiplexer model, request blocking occurs at a link whenever a number of connections, larger than the speedup of the link, contend for the same slot. Since contention at a link may be affected by blocking at other links in the network (e.g., upstream links), we define contention as follows.

**Definition 2 (Contention).** Let  $\tau$  be a time slot on some link in the network. Contention at  $\tau$ , under a regime  $G$  (a network-wide routing and bandwidth allocation scenario), denoted  $\chi_\tau^G$ , is a random variable representing the number of requests for slot  $\tau$  under  $G$ .

The probability of blocking at a link is the probability of contention at any slot exceeding the link speedup. Then, the probability of blocking at a link is maximum when contention at every slot, as a random variable, is (stochastically) maximum given the prescribed link load:

**Definition 3** (*Regimes of Maximal Contention*) Let  $\mathcal{G}$  be the family of regimes that satisfy a prescribed load at link  $l$ . A regime  $G^* \in \mathcal{G}$  is said to result in maximal contention at link  $l$  if for each slot  $\tau$  on  $l$ ,  $\Pr\{\chi_\tau^{G^*} > \kappa\} \geq \Pr\{\chi_\tau^G > \kappa\}$  for all  $G \in \mathcal{G}$  and  $\kappa \geq 1$ . Equivalently, for each  $G \in \mathcal{G}$ ,  $G^*$  is said to result in higher contention at  $l$  compared to  $G$ .

The above definitions provide a convenient language for reasoning about conditions that maximize the expected blocking rate at a link without any assumptions about the scheduling algorithm used (except work conservation). It is easy to see that if a regime results in maximal contention at link  $l$ , then it not only maximizes the probability of blocking at any given slot (irrespective of speedup), but also maximizes the expected number of blocking events at that slot. That is, it maximizes the expected blocking rate (as a fraction of total transmissions) at the link. Furthermore, since the expected blocking rate must equal the average of the expected blocking rate of all connections at  $l$  (each weighted by the connection's normalized contribution to the load), an increase in the expected blocking rate at  $l$  may increase but never decrease the expected blocking rate of each connection. Thus, in bounding the blocking rate of a connection under any scheduling algorithm, we should consider only regimes that maximize contention at every link it traverses.

Back to our analysis of BATCH. To completely characterize contention at a link, it is sufficient to study contention over a single link period. This follows directly from Property 3 given the definition of a link period (Definition 1). Furthermore, observe that contention is maximal at a link  $l$  only if the time slot requests from different connections are statistically independent. Otherwise by (M3), a subset of connections must share at least one upstream link, hence are less likely to contend for the same slot.<sup>8</sup> When the statistical independence of time slot requests is satisfied, we simply say that the connections sharing  $l$  are statistically independent. Building on these two observations, the following theorem states that a regime results in maximal contention at a link iff all connections routed through the link are statistically independent, have bandwidth allocation equal to  $\rho_0$  (the smallest possible allocation), and do not experience blocking at upstream links.

**Theorem 1** Let  $\mathcal{G}$  be the family of regimes that satisfy a prescribed load at link  $l$  such that all connections sharing  $l$  under any regime in  $\mathcal{G}$  are statistically independent. A regime  $G^* \in \mathcal{G}$  results in maximal contention at link  $l$  iff for every connection  $c$  routed through  $l$  under  $G^*$ , the pattern of slots requested by  $c$  on  $l$  is the pattern defined by  $\alpha_{\rho_0, \mu_l}$ .

**Proof:** See Appendix. ■

The theorem formally establishes the intuitive observation in [16] that the probability of blocking in a multiplexer of periodic packet streams is maximal at a given load, only when

<sup>8</sup>For instance, these connections cannot contend for the same slot at  $l$  if the upstream link has speedup 1.

all connections have minimum transmission rates. That is, when the number of statistically independent periodic sources contributing to the load is maximum. Although at first glance this appears to be an obvious result, it becomes less so under careful consideration. On one hand, if each connection requests only 1 slot per link period (i.e., has minimum bandwidth allocation), there is a better chance that the link receives a large number of requests for the same slot. This is because requests by the same connection must be for different slots. On the other hand, when a connection requests multiple slots per link period, because of periodicity, each request falls into a smaller interval of slots within the link period. As a result, the probability that a particular slot is chosen by the connection is higher. This raises the possibility that even though a fewer number of independent connections are contributing to the load in the second case, contention at a slot may end up being higher, in the formal sense.

The theorem implies that the blocking rate of a tagged connection traversing a sequence of links is maximal given the load at the links only if the following conditions are satisfied:

- (1) it competes at each link against a different set of connections (which we call, the “background connections” at the link), and at every link,
- (2) the background connections do not experience request blocking along the upstream portions of their paths, and,
- (3) the tagged connection contributes a small portion of the load at every link so that the blocking it experiences upstream does not significantly improve the probability of blocking of its surviving requests.

Together (1), (2) and (3) imply that to obtain an upper-bound on the blocking rate along the path of the tagged connection, each link can be studied in isolation, thus leading to a product-form expression for the overall request blocking probability along the tagged connection's path.

**Remarks.** The proof of Theorem 1 uses the properties of the (staircase) request pattern of BATCH. It is otherwise oblivious to all other aspects of the scheduling algorithm. Therefore, the theorem applies to networks using any work-conserving scheduling rule provided that the pattern of time-slot requests by all connections follow the periodic staircase pattern.

When the EA rule is used instead of BATCH in the reservation-based framework, each reservation packet carries only one time slot request. If the requests are generated according to the staircase pattern, Theorem 1 implies that the analysis of EA in Section 3 is actually a worst-case analysis of the request blocking probability along a path of identical links. The same applies when EA is used in the context of a traditional packet switching network (without reservations) provided that packets entering the network are slot-sized, and that the traffic of each connection is shaped at the ingress into

a periodic stream.<sup>9</sup>

In summary, the results in this section provide a framework for the analysis of BATCH and comparing its performance to the EA scheduling rule. We analyze the link blocking performance under BATCH in the next section.

## 6 Blocking at a Link

In this section, we study blocking within a reservation epoch from an arbitrarily tagged connection  $c$ , at a link  $l$ . We derive bounds on the probability of blocking of individual requests in the batch covering the reservation epoch. The bounds are exact expressions for the blocking probabilities under the conditions specified by Theorem 1. They are therefore tight.

### 6.1 Preliminaries

Consider a particular reservation epoch of the tagged connection  $c$  as a reference epoch. The reference epoch spans  $n$  periods at link  $l$  numbered  $1, 2, \dots, n$ . Let  $N$  be the number of connections sharing link  $l$ . Since all reservation epochs have the same duration, the reference epoch overlaps with either one epoch (complete overlap), or two reservation epochs from each of the remaining  $N - 1$  connections. Complete overlap of reservation epochs occurs when the respective connections start (and end) their epochs at the same link slot.

Due to phase randomization (Section 4.1.2), reservation epochs from each of  $N - 1$  connections overlap with  $c$ 's reference epoch at a uniform random phase. That is, each connection is equally likely to end their current epoch and start a new one at any slot within the reference epoch.

Suppose that we can ignore upstream processing delay encountered by a reservation packet, which typically depend on the length of the upstream portion of its path. Then reservation packets arrive at link  $l$  in the same temporal order of the start of the corresponding reservation epochs. Specifically, a reservation packet is received at the link ahead of the start of the corresponding reservation epoch by duration equal to the horizon (Section 4). If, due to upstream processing delays, reservation packets are not received at the link in the proper order, we say that the order of reservations is violated.

Let  $\delta$  be the maximum upstream processing facing any connection, expressed in time slots. Then  $n$  can be chosen large enough that, with arbitrarily high probability, the order of reservations is not violated. Suppose that  $c$ 's reservation packet is received at link  $l$ , delayed by  $\delta$  slots. The probability that the order of reservations is violated is at most the probability that another connection starts a reservation epoch within  $\delta$  slots from the start of  $c$ 's epoch. Due to phase randomization, this probability is  $\frac{\delta}{nT_{0,l}}$ . Thus, for a fixed time slot duration and link period, this probability can be made ar-

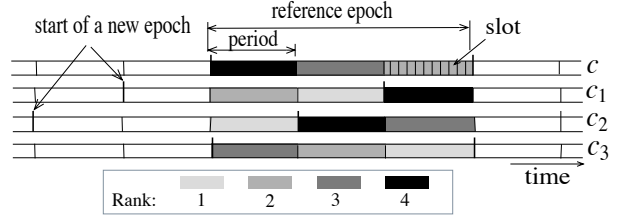


Figure 5: Illustration of connection ranks.

bitrarily small by the choice of  $n$ . Moreover, the value of  $n$  does not have to be very large as the time-slot duration (equivalently, data frame size in bits) can be chosen so that  $\delta$  is a small fraction of a time slot. In the following analysis, we assume that the probability of violating the order of reservations is negligibly small.

For notational convenience, we consider epochs starting within the slots of a period within  $c$ 's epoch, to have started at that period's leading boundary. This implies that each period of the reference epoch belongs to exactly one epoch of each connection, hence covered by one reservation packet (batch of requests) from each of them.

Define the rank of a connection during a period as the order of scheduling of its batch of requests relative to the  $N$  request batches covering the period. Recall that request batches are processed, without interleaving, in the order of reception of the corresponding reservation packets at the link's interface. We assume that reservation packets are received in the same temporal order of the starting slots of the corresponding epochs.

Let  $R_1, R_2, \dots, R_n$  be a sequence of random variables denoting the ranks of connection  $c$  during the periods of the reference epoch. In every realization,  $R_1, \dots, R_n$  is a decreasing sequence since within  $c$ 's epoch, each of the connections competing with  $c$  has to end an epoch and start a new one, during which it succeeds connection  $c$  in rank.

The concept of ranks is illustrated in Figure 5. The diagram shows the reservation epoch of connection  $c$  under consideration (the reference epoch) along with the overlapping epochs of the connections competing with  $c$ , namely  $c_1, c_2$ , and  $c_3$ . The duration of a reservation epoch spans  $n = 3$  link periods. Connection  $c$ 's reservation epoch (at the top) overlaps with two epochs from each of  $c_1$  and  $c_2$ , and one epoch of  $c_3$ . The figure shows the rank of each connection during the periods of the reference epoch. Connection  $c$  succeeds  $c_1$  in rank during the first two periods of its reservation epoch and precedes it during the third. This is because the first two periods of  $c$ 's epoch overlap with an epoch of  $c_1$  that had started earlier, whereas the third period overlaps with the first period of a new epoch of  $c_1$ . In this example connection  $c$  ranks fourth, third, and second (in that order) in the three periods spanned by its epoch. Connection  $c_3$  precedes  $c_1$  throughout their completely overlapping epochs. Since the rank during a

<sup>9</sup>Note that the performance of EA without ingress shaping would be worse due to burstiness of packet arrivals.

period is defined by the order of processing reservation packets covering that period,  $c$ 's requests in period  $i$  of the reference epoch, can be blocked only due to contention caused by connections that precedes it in rank. We start by assuming that connection  $c$  requests one slot per link period: define the random variables  $X_{1,l}, X_{2,l}, \dots, X_{n,l}$  as

$$X_{i,l} \triangleq \begin{cases} 1 & c\text{'s request within period } i \text{ is blocked} \\ 0 & \text{otherwise.} \end{cases}$$

Due to periodicity in the generation of reservation packets at the ingress routers, the order of reception of reservations packets at link  $l$  remains unchanged in every epoch preceding or succeeding the reference epoch. Thus the analysis of  $X_{i,l}$ ,  $i = 1, \dots, n$  applies to every reservation epoch of connection  $c$ . Let  $X_l = \sum_{i=1}^n X_{i,l}$ . Then the worst-case expected blocking rate that can be experienced by connection  $c$  at link  $l$  is given by:

$$\mathbb{E}[X_l/n] = \frac{1}{n} \sum_{i=1}^n \Pr\{X_{i,l} = 1\}, \quad (2)$$

where,

$$\Pr\{X_{i,l} = 1\} = \sum_{k=1}^N \Pr\{X_{i,l} = 1 | R_i = k\} \Pr\{R_i = k\}. \quad (3)$$

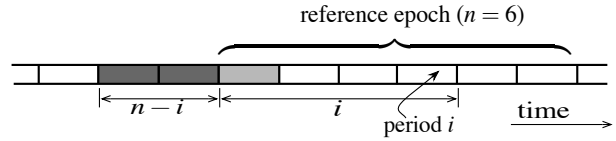
During period  $i > 1$ ,  $c$  faces blocking due to contention among a subset of the connections that preceded it in rank during period  $i - 1$ . This indicates that  $\Pr\{X_{i,l} = 1\}$  should be decreasing over the periods of the epoch. A proof is given in Section 8.

It is worth noting that Equation (2) is an upper bound on the  $c$ 's expected blocking rate when it requests multiple slots per link period. This is because, the event that all of  $c$ 's requests within period  $i$  are blocked, occurs with probability no greater than  $\Pr\{X_{i,l} = 1\}$ .

In the following subsections, we obtain expressions for the distribution of ranks and bounds on the blocking probabilities, then use them to establish the properties of the transmission-scheduling algorithm.

## 6.2 Distribution of ranks

For  $k = 1, \dots, N$ ,  $\Pr\{R_i = k\}$ , is the probability that for any  $r \in \{0, 1, \dots, k - 1\}$ ,  $(k - 1) - r$  connections (out of the  $N - 1$  connections competing with  $c$ ) start new epochs within the sequence of consecutive periods of length  $n - i$  that ends immediately before the first period of  $c$ 's epoch (the reference epoch), and at least  $r$  connections (in addition to  $c$ ) start new epochs at the beginning of the reference epoch, with  $c$  always ranking the  $i + 1$ st among them (see Figure 6). Thus, for



**Figure 6:** The Rank of connection  $c$  during period  $i$ . Connections that start epochs within the dark-shaded periods precede  $c$  in rank during  $i$ . Those are in addition to connections starting new epochs during the first period of  $c$ 's epoch (light-shaded) and whose reservation packets are received before  $c$ 's packet.

$1 < i < n$ :

$$\Pr\{R_i = k\} = \sum_{r=0}^{k-1} \binom{N-1}{k-r-1} \left(1 - \frac{i}{n}\right)^{k-r-1} \cdot \sum_{j=r}^{N-k+r} \frac{1}{j+1} \binom{N-k+r}{j} \left(\frac{1}{n}\right)^j \left(\frac{i-1}{n}\right)^{N-k+r-j}, \quad (4)$$

where the factor  $\frac{1}{j+1}$  represents the probability that  $c$ 's reservation packet is the  $i + 1$ st to be processed among the  $j$  packets that belong to the connections whose epochs completely overlap with  $c$ 's, which is  $\frac{j!}{(j+1)!} = \frac{1}{j+1}$ .

For  $i = 1$ , all of the  $N - 1$  connections have to start new epochs within the  $n$  periods ending with and including the first period of the reference epoch. Then,

$$\Pr\{R_1 = k\} = \sum_{r=0}^{k-1} \frac{1}{N-k+r+1} \binom{N-1}{k-r-1} \cdot \left(1 - \frac{1}{n}\right)^{k-r-1} \left(\frac{1}{n}\right)^{N-k+r}. \quad (5)$$

Similarly, for  $i = n$ , all connections have to start new epochs within the reference epoch. Then:

$$\Pr\{R_n = k\} = \sum_{j=k-1}^{N-1} \frac{1}{j+1} \binom{N-1}{j} \left(\frac{1}{n}\right)^j \left(1 - \frac{1}{n}\right)^{N-1-j}. \quad (6)$$

Recall that any realization of the sequence  $R_1, R_2, \dots, R_n$  is monotonically decreasing. Whereas  $R_n$  is at most the number of connections that started new epochs at the beginning of the reference epoch,  $R_1$  is all of the  $N - 1$  connections except those starting new epochs at the first period of the reference epoch, and whose reservation packets happen to be received after  $c$ 's packet.

## 6.3 Rank-based blocking probabilities

We now obtain an expression for the conditional probability  $\Pr\{X_{i,l} = 1 | R_i = k\}$ . Let the size of a link period on  $l$  be  $M$  slots, and that  $l$  has speedup  $s$  requests (frames) per slot. Then connection  $c$ 's request during period  $i$  is blocked iff at least  $s$  connections out of the  $k - 1$  preceding  $c$  in rank during

that period request the same slot as  $c$ . Thus, we have:

$$\Pr\{X_{i,l} = 1 | R_i = k\} = \sum_{j=s}^{k-1} \binom{k-1}{j} \left(\frac{1}{M}\right)^j \left(1 - \frac{1}{M}\right)^{k-1-j}. \quad (7)$$

The above equation is obtained by observing that, due to uniform phase randomization, each connection is equally likely to request any of the  $M$  slots in period  $i$ .

## 7 Multihop Paths

In this section, we provide expressions for the expectation of blocking rate experienced by an arbitrarily tagged connection, and its tail probability. The expressions result directly from the analysis in previous sections, under the conditions of Theorem 1.

### Expected blocking rate bound

Since connection bandwidth allocations are integral multiples of  $\rho_0$  (assumption (M2)), the number of link periods spanned by a reservation epoch at every link is equal to the minimum batch size (Section 4). Consider the case where the minimum batch size is chosen such that a reservation epoch spans  $n$  periods on each link, and let  $\mathcal{P}_c$  denote the path of the tagged connection,  $c$ . Furthermore, let  $X_1, X_2, \dots, X_n$  indicate whether the request in period  $i$  of an epoch of connection  $c$  has been blocked at any link along the path.

Under the conditions of Theorem 1, the blocking probability of a request  $\Pr\{X_i = 1\}$  has the following product form:

$$1 - \Pr\{X_i = 1\} = \prod_{l \in \mathcal{P}_c} (1 - \Pr\{X_{i,l} = 1\}), \quad (8)$$

Let  $X$  be the number of blocked requests in the reservation epoch, i.e.,  $X \triangleq \sum_{i=1}^n X_i$ . Then  $\mathbb{E}[X/n]$  is the expected blocking rate for connection  $c$ .

$$\mathbb{E}[X/n] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[X_i] = \frac{1}{n} \sum_{i=1}^n \Pr\{X_i = 1\} \quad (9)$$

Substituting from (8), we get

$$\mathbb{E}[X/n] = 1 - \frac{1}{n} \sum_{i=1}^n \prod_{l \in \mathcal{P}_c} (1 - \Pr\{X_{i,l} = 1\}). \quad (10)$$

### Tail probability bound

Consider the sequence  $X_1, \dots, X_n$  defined in the previous section. Since the pattern of requested slots by each connection is periodic at every hop, then for  $i = 2, \dots, n$ ,  $\Pr\{X_i = 1\} = \Pr\{X_i = 1 | X_{i-1} = 1\} \Pr\{X_{i-1} = 1\}$ . In other words,  $X_i = 0$  implies  $X_j = 0$  for  $j = i+1, \dots, n$ . This immediately produces a bound on the tail of the blocking rate distribution:  $\Pr\{X/n \geq i/n\} = \Pr\{X_i = 1\}$ , where  $X = X_1 + X_2 + \dots + X_n$  as defined above.

## 8 Qualitative Results

In this section, we establish some qualitative properties of BATCH regarding correlation of scheduling decisions at different links, and fairness in apportioning blocking events among connections sharing each link.

### 8.1 Correlation of scheduling decisions

We say that BATCH is correct if, under worst-case conditions (of Theorem 1) a fraction of requests from each connection are consistently subject to blocking probability smaller than the expected blocking rate at every link they traverse. Correctness is defined in terms of worst-case routing and bandwidth allocation scenarios because the bounds corresponding to these conditions are the ones used for deriving link load and path length constraints on connection routing. These constraints determine the network's connection-carrying capacity.

If correct, BATCH would result in better utilization–path length tradeoffs compared to algorithms, such as EA, where all request traversing a link are subject to identical blocking probability, equal to the expected link blocking rate. From (8), requests subject to lower blocking probability at every hop are likely to survive a larger number of hops.

Let  $g(k) = \Pr\{X_{i,l} = 1 | R_i = k\}$  for some period  $i \in \{1, \dots, n\}$  on link  $l$ . Then the following result is obvious from (7):

**Lemma 1** *The function  $g$  is independent of  $i$ , and is an increasing function of  $k$ .*

To prove the correctness of the algorithm we need to define the following strict stochastic order relation.

**Definition 4** *Consider two discrete random variables  $Y_1$  and  $Y_2$  with finite support  $S \subset \mathbb{Z}$ , and let  $m = \max S$ . We say that  $Y_1$  is (stochastically) strictly larger than  $Y_2$  (denoted  $Y_1 >_{st} Y_2$ ) if for all  $a \in S \setminus \{m\}$ ,  $\Pr\{Y_1 > a\} > \Pr\{Y_2 > a\}$ .*

**Lemma 2**  $Y_1 >_{st} Y_2 \Rightarrow \mathbb{E}[f(Y_1)] > \mathbb{E}[f(Y_2)]$  for all increasing functions  $f$ .

**Proof:** Similar to [17, Proposition 9.1.2]. ■

Now we state our correctness result:

**Theorem 2**  $(\Pr\{X_{i,l} = 1\}, i = 1, \dots, n)$ , is a strictly decreasing sequence.

**Proof:** We sketch the proof. According to (3), we need to show that  $\mathbb{E}[g(R_i)] > \mathbb{E}[g(R_{i+1})]$  for  $i = 1, \dots, n-1$ . Using a coupling argument, we can show that

$$R_n >_{st} R_{n-1} >_{st} \dots >_{st} R_1.$$

Then, by Lemma 2, for any increasing function  $f$ , and  $i = 1, \dots, n - 1$ :

$$\sum_{k=1}^N f(k) \Pr\{R_i = k\} > \sum_{k=1}^N f(k) \Pr\{R_{i+1} = k\}. \quad (11)$$

Since  $g(k) \triangleq \Pr\{X_{i,l} = 1 | R_i = k\}$  is independent of the period index  $i$ , and is an increasing function of  $k$  (Lemma 1), we can substitute  $g$  for  $f$  in the equation above. Thus completing the proof. ■

Theorem 2 implies that a subset of requests from (each epoch of) each connection are consistently subject to blocking probability smaller than the expected link blocking rate, at every link.

## 8.2 Fairness of BATCH

We say that a scheduling algorithm is fair if, at every link, all connections sharing the link have identical tight bounds on the expected blocking rate at the link. Note that without the tightness requirement, a common upper bound may not be indicative of relative worst-case performance of different connections. The following theorem establishes the fairness of BATCH.

**Theorem 3** BATCH is fair.

**Proof:** Substituting from (3) into (2) we get the following bound on the expected blocking rate of an arbitrarily tagged connection at some link  $l$ :

$$\mathbb{E}[X_l/n] = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^N \Pr\{X_{i,l} = 1 | R_i = k\} \Pr\{R_i = k\} \quad (12)$$

The bound is tight since (7) gives an exact expression for  $\Pr\{X_{i,l} = 1 | R_i = k\}$  when the conditions of Theorem 1 are satisfied at  $l$ . Furthermore, because of the choice of the tagged connection, the bound applies to all connections routed through  $l$ . ■

The fairness result implies that any quantitative improvement in the expected (path) blocking rate bounds for connections traversing a large number of hops (compared to EA) does not come by improving their expected blocking rate bounds at any link while worsening the bounds for other connections (e.g., those routed over shorter paths).

**Remarks.** Recall that the characterization of  $R_1, \dots, R_n$  was obtained in under the assumption that  $n$  is large enough that the order of reservations is almost never violated (more precisely, the probability of violation is negligible). When  $n$  is too small, the bounds in Section 7 and the qualitative results presented in this section do not apply: the processing delays experienced by a reservation packet are large enough that some connection experiencing large upstream processing

delays is subject to higher request blocking probability compared to other connections. In such case, the best bound on the blocking probability is (1), which states that the blocking probability of a request at a given link is bounded by the probability that a number of other connections at least equal to the link speedup request the same link slot. That is, when  $n$  is too small, the performance of BATCH is equivalent to EA with respect to the expected blocking rate. As discussed in Section 6, both  $n$  and the time slot duration should be chosen so that the probability of violating the order of reservations is negligible.

## 9 Numerical Results

In this section, we evaluate the effectiveness of BATCH in improving the traffic-carrying capacity of bufferless networks. This is in comparison to the “natural” bufferless scheduling rule EA. As in Section 2, we use the tradeoff between the link utilization and path length to guarantee a desired bound on the expected blocking rate of a connection. We refer to such tradeoffs as “ $U$ - $P$  tradeoffs”.

We consider a network of identical links under the conditions of Theorem 1 where each connection requests only one slot per link period. The links are identical in the sense of having equal capacity and speedup. The capacity is expressed as number of slots per link period.

Under normal operating conditions, we are interested in the  $U$ - $P$  tradeoffs to achieve an expected blocking rate in the interval  $(0, 0.1]$ , when network link can be shared by up to few hundred independent connections (ingress-egress aggregates) and the network diameter extends to 30 hops. Recall that if the expected blocking rate bound is 0.1, the expected capacity of the bandwidth tunnel allocated to a connection is at least 90% of the nominal capacity.

We know from Section 2 that, at large values of speedup ( $s \geq 100$ ), EA results in good  $U$ - $P$  tradeoffs. We also know that large speedup values may not be cost-effective. Our numerical results indicate that for small values of link speedup (e.g., speedup  $s \leq 10$ ) both EA and BATCH are unable to provide the desired  $U$ - $P$  tradeoffs. However, for speedups in the range  $10 < s < 100$ , BATCH provides substantial improvements over EA.

Improvements in  $U$ - $P$  tradeoffs due to BATCH can be seen in the plots of Figures 7 and 8. The plots are obtained using the following settings: the link capacity is 128 data frames per link period, and the number of slots per link period is  $M = 128/s$ , where  $s$  is the link speedup. Each connection requests only 1 slot every link period. Thus, at a given link load, the number of independent connections competing at the link is fixed irrespective of the speedup.

The leftmost plot of Figure 7, shows that, at 60% link utilization (load), EA can provide an expected blocking rate of 0.1

for connections traversing at most 10 hops. This is compared to BATCH with  $n = 20$  (rightmost plot) which can provide the same guarantee at 70% link utilization for connections traversing more than 25 hops. For an expected blocking rate guarantee of 0.02 BATCH can support path lengths up to 25 hops at utilization 60%, compared to less than 5 hops for EA under the same conditions.

Providing the same expected blocking rate guarantee as EA over longer paths, but without significant improvements in maximum link utilization, effectively improves the network's connection-carrying capacity by allowing the network operators to route traffic over paths that are unusable with EA. On the other hand, even modest improvement in maximum link utilization (sufficient to carry an extra connection) may translate into an improvement in connection-carrying capacity that is linear in the number of hops. As a simple example, on average each link may be able to carry an extra half connection when the average path length is 2 hops.

Comparing the tradeoffs under BATCH with  $n = 10$  and  $n = 20$  shows that the tradeoffs improve with increasing  $n$ . This is because a larger number of requests in each epoch face small blocking probability at each link (the rank remains small for a larger number of periods). Extremely large values of  $n$  are not desirable, however, from a packet delay perspective: There may be a large number of consecutively blocked requests at the beginning of each epoch resulting in large packet delay at the ingress.

Figure 8, shows the effect of higher link speedup on the tradeoffs improvements provided by BATCH. Whereas, the leftmost plot shows that EA cannot support an expected blocking rate guarantee of 0.02 at 80% link utilization (even over 1-hop paths), BATCH can provide this guarantee at the same load over paths up to 20 hops. By comparing the rightmost plot ( $s = 50, n = 20$ ) with the EA at  $s = 80$  (not shown), we found that they provide almost identical tradeoffs. This leads us to conclude that BATCH can provide the same tradeoffs as EA at up to 37.5% saving in speedup.

Under the conditions of Theorem 1, increasing the link capacity (in frames per link period) while maintaining the same load implies increasing the number of independent connections contending for link slots. We therefore expect the  $U-P$  tradeoffs to deteriorate. The tradeoffs in Figures 9 are obtained by doubling the link capacities compared to the results in Figure 7. That is the link capacities are taken to be 256 frames/period. Comparing both figures shows that doubling the link capacity results in slightly worse tradeoffs. Still, BATCH provides significant tradeoff improvements over EA.

## 10 Concluding Remarks

Now we give some remarks regarding the complexity of BATCH, and discuss possible avenues for further research. It is easy to see that work performed by the ingress to generate

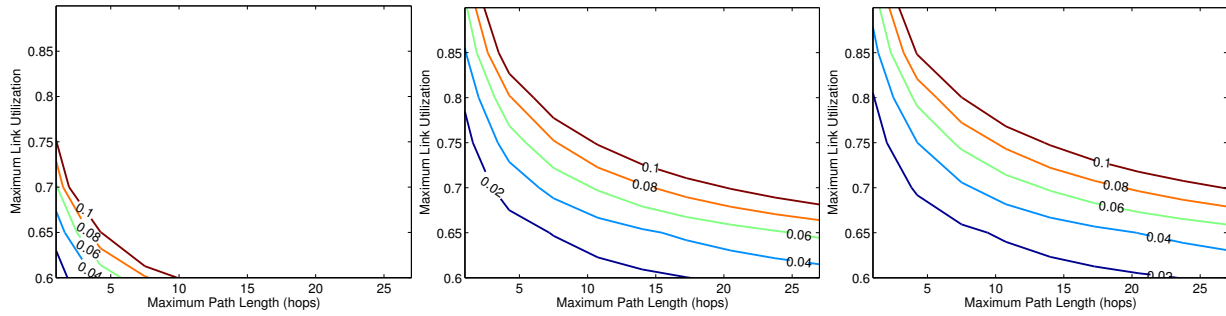
a time slot request is  $O(1)$ . The same applies to work performed by a link to schedule a transmission request. At any given time, a link may receive transmission request batches that together span the duration of up to two (future) reservation epochs. Since the link has to keep track of the availability of link slots, the space complexity of the algorithm is twice the size of the reservation epoch, hence linear in the number of link periods per epoch. The exact storage requirements depend on the size of a link period in slots (or equivalently, the bandwidth allocation granularity,  $\rho_0$ ). Each slot is represented by a number of bits proportional to the link speedup. This number should be large enough to specify the input ports that will offer data frames (packets) to the output link during the slot (i.e., log the sum of link speedups). The schedule of reservation is used to configure the switch fabric to transfer arriving packets at the input to the proper output links. It should be noted that the storage requirement typically ranges from fraction of one Mbit to few Mbits [3]. This is not a large burden on routers, especially that memory used by the scheduler need not be as fast as packet buffers, hence are not subject to the same capacity limitations.

BATCH can be extended in straightforward manner to networks where links are equipped with limited contention-resolution buffering capacity [3]. As in [14], we limited our presentation to the bufferless case for reasons of analytical tractability. Simulation results not reported in this paper indicate that BATCH improves the path-length vs. utilization tradeoffs in networks with limited buffers and link speedup = 1 compared to the FCFS scheduling rule.

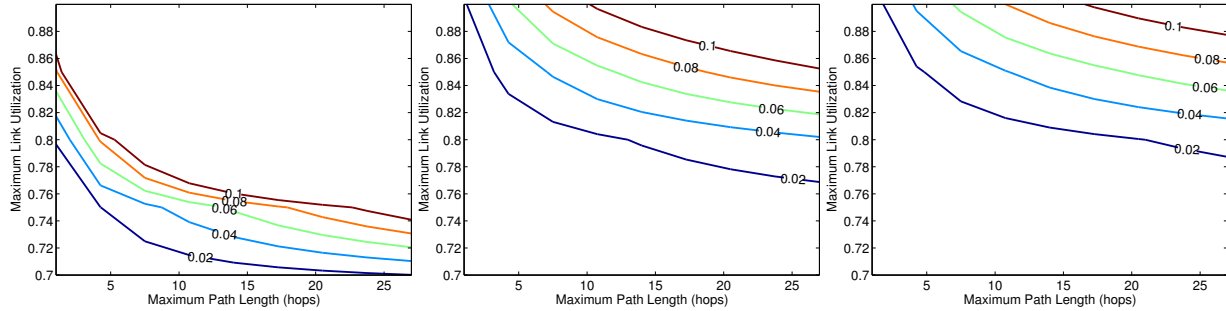
The underlying idea in BATCH is that of **Rolling Priorities**. From the point of view of a connection, time is divided into fixed-size epochs. Packet transmitted by the connection toward the end of an epoch are subject to a small loss (blocking) probability at each link. In other words, packets at the beginning of an epoch are treated as if they have low scheduling priority at each link, while packets towards the end are treated as if they have high priority. At any link, the overlap of epochs from connections sharing the link should minimize the chance that higher-priority packets from different connections arrive simultaneously at the link. This is required when the expected blocking rate at the link is high, since the low blocking probability for higher-priority packets must be achieved at the expense of low priority ones. The challenge is to implement Rolling Priorities without limiting network scalability. For example, without explicit coordination of scheduling decisions among different links. BATCH avoids contention among high priority requests using randomization.

## References

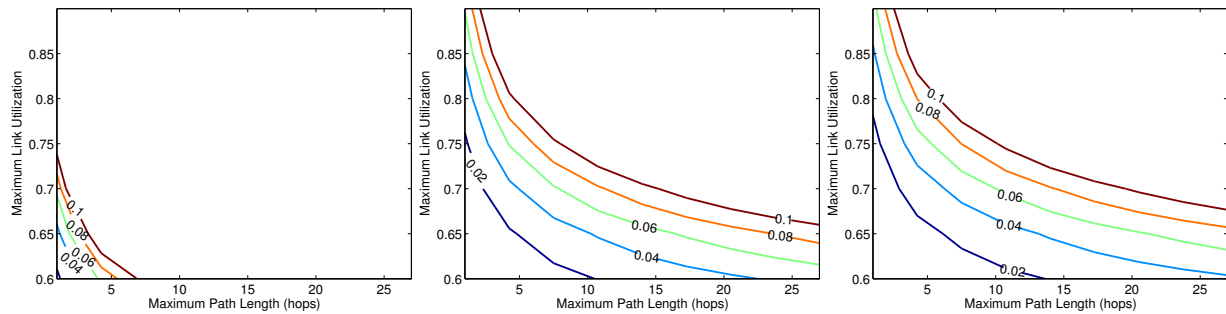
- [1] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *ACM SIGCOMM '04*, August /September 2004.
- [2] Y. Chen, C. Qiao, and X. Yu. Optical burst switching: A new area in optical networking research. *IEEE Network*, 18(3):16–23, May 2004.
- [3] M. Elhaddad, R. Melhem, and T. Znati. Sup-



**Figure 7:** Link utilization versus path length ( $U$ - $P$ ) tradeoffs for speedup  $s = 20$  and link capacity 128 slots/period. Left: EA, middle:  $n = 10$ , right:  $n = 20$ . Each contour is the locus of a constant expected blocking rate in the  $U$ - $P$  plane.



**Figure 8:**  $U$ - $P$  tradeoffs for speedup  $s = 50$ . Except for the difference in speedup, the plots are obtained under the same settings of Figure 7.



**Figure 9:**  $U$ - $P$  tradeoffs for speedup  $s = 20$  and link capacity 256 slots/period. Except for the difference in link capacity, the plots are obtained under the same settings of Figure 7.

porting bandwidth guarantees in buffer-limited networks. Technical report, University of Pittsburgh. <http://www.cs.pitt.edu/~elhaddad/BCN/>.

[4] M. Elhaddad, R. Melhem, and T. Znati. Decoupling packet loss from blocking in Proactive Reservation-based Switching. In *Proceedings of Broadband Networks, Optical Networks Symposium*, 2004. <http://www.cs.pitt.edu/~elhaddad/BCN/>.

[5] S. Floyd. HighSpeed TCP for large congestion windows. Technical report, IETF RFC 3689 (experimental), 2003.

[6] S. Floyd and V. Jacobson. Traffic phase effects in packet-switched gateways. *Journal of Internetworking: Practice and Experience*, 3(3):115–156, September, 1992.

[7] T. Kelly. Scalable TCP: improving performance in highspeed wide-area networks. *SIGCOMM Computer Communication Review*, April 2003.

[8] H. S. Kim and N. B. Shroff. Loss probability calculations and

asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Trans. Netw.*, 9(6):755–768, 2001.

[9] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *ACM Sigmetrics*, pages 128–139, June 1992.

[10] C. Li and E. Knightly. Coordinated multihop scheduling: A framework for end-to-end services. *IEEE/ACM Trans. Netw.*, 10(6), December 2002.

[11] M. Parulekar and A. M. Makowski. Tail probabilities for a multiplexer with self-similar traffic. In *IEEE INFOCOM*, 1996.

[12] A. Pattavina. Architectures and performance of optical packet switching nodes for IP networks. *Journal of Lightwave Technology*, 23(3):1023–1032, March 2005.

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

[14] M. Reisslein, K. W. Ross, and S. Rajagopal. A framework for

guaranteeing statistical QoS. *IEEE/ACM Trans. Netw.*, 10(1):27–42, 2002.

[15] J. Roberts, U. Mocci, and J. V. (Eds.). Broadband network traffic: Performance evaluation and design of broadband multiservice networks. In *Final report of Action COST 242*, volume 1155 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[16] 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.

[17] S. M. Ross. *Stochastic Processes*. John Wiley & Sons, Inc., second edition, 1996.

[18] G. Shrimali, I. Keslassy, and N. McKeown. Designing packet buffers with statistical guarantees. In *IEEE Hot Interconnects XII*, Stanford, CA, 2004.

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

[20] 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.

## Appendix: Proof of Theorem 1

### Necessity proof

First we establish the intuitive result that a regime  $G \in \mathcal{G}$  produces maximal contention at  $l$  only if the time-slot requests on  $l$  under  $G$  face zero blocking probability at upstream links. In other words, only if the request pattern of each connection  $c$  traversing  $l$  under  $G$  is defined by  $\alpha_{\rho_c, \mu_l}$ .<sup>10</sup> Suppose that  $G$  is a regime of maximal contention at  $l$ . To reach contradiction, we construct a regime  $G' \in \mathcal{G}$  that results in *higher contention* at  $l$ ; that is  $\Pr\{\chi_\tau^{G'} > \kappa\} \geq \Pr\{\chi_\tau^G > \kappa\}$  for all  $\kappa \geq 1$ . Let  $\mathcal{C}_l^G$  be the set of connections sharing  $l$  under a regime  $G \in \mathcal{G}$ . Since a regime is a global routing and bandwidth allocation scenario, there may be other regimes in  $\mathcal{G}$  that route the same set of connections through  $l$ , with the same bandwidth allocations, but differ from  $G$  in the set of connections routed through some other link(s) or their bandwidth allocations. Let this subset of  $\mathcal{G}$  (including  $G$  itself) be labeled  $\mathcal{G}_G$ . Consider any sequence of consecutive of length  $M = T_{0,l}$  on  $l$ , numbered  $0, 1, \dots, M - 1$ , and suppose that a request for slot  $\tau \in \{1, \dots, M\}$  from connection  $c$  has upstream blocking probability  $p > 0$  under  $G$ . Then we can write for all  $\kappa \geq 1$ :

$$\begin{aligned} \Pr\{\chi_\tau^{G'} > \kappa\} &= p \cdot \Pr\{\chi_\tau^G > \kappa \mid c\text{'s request for } \tau \text{ is blocked}\} \\ &\quad + (1 - p) \cdot \Pr\{\chi_\tau^G > \kappa \mid c\text{'s request for } \tau \text{ is not blocked}\}. \end{aligned}$$

Consider some  $G' \in \mathcal{G}_G$  that is identical to  $G$  except that  $c$ 's request for  $\tau$  has zero upstream blocking probability (e.g.,  $c$  does not share upstream links with any other connections). For  $1 \leq \kappa \leq |\mathcal{C}_l^G|$ , we have:

$$\begin{aligned} \Pr\{\chi_\tau^{G'} > \kappa\} &= \Pr\{\chi_\tau^G > \kappa \mid c\text{'s request for } \tau \text{ is not blocked}\} \\ &> \Pr\{\chi_\tau^G > \kappa\}, \end{aligned}$$

which contradicts  $G$  being a regime of maximal contention at  $l$ .

<sup>10</sup>Recall that the request pattern of a connection  $c$  is defined by  $\alpha_{\rho_c, \mu_l}$  if for some starting slot  $\tau_0$  on  $l$ , the pattern matches the one generated by  $\alpha_{\rho_c, \mu_l}$  using the ingress rule (Section 4.1.1).

Now we argue that  $G \in \mathcal{G}$  results in a regime of maximal contention at  $l$  only if  $\rho_c = \rho_0$  for every  $c \in \mathcal{C}_l^G$ . Suppose that the pattern of requested slots by each connection  $c \in \mathcal{C}_l^G$  is defined by the pattern  $\alpha_{\rho_c, \mu_l}$  to satisfy the first necessary condition. By assumption (M2), if  $\rho_c > \rho_0$  for some connection  $c \in \mathcal{C}_l^G$ , then  $c$  requests an integral number of requests greater than 1 every link period. Note however that this number cannot exceed  $M$  irrespective of the link speedup.

Let  $N = |\mathcal{C}_l^G|$  and suppose that connections are arranged in non-decreasing order of their number of requests per period so that  $r_1 \leq r_2 \leq \dots \leq r_N$  where  $r_j$  denotes the number of requests per period by the  $j$ th connection. Further, let the slots of a particular link period be numbered  $0, 1, \dots, M - 1$ . Since connection request patterns are periodic, the  $i$ th request from the  $j$ th connection,  $i = 0, \dots, r_j - 1$ , must be for a slot in the segment (interval) of the slots  $\left[ i \frac{M}{r_j}, (i+1) \frac{M}{r_j} \right]$  within the period.<sup>11</sup> Furthermore, the choice of slot in a segment determines the choice in all segments. Therefore, if  $r_j > 1$ , the  $r_j$  requests are not statistically independent (these requests cannot conflict). Uniform phase randomization implies that connection  $j$  is equally likely to choose any slot within a segment.

Let  $n_m$  denote the number of requests up to and including requests from the  $m$ th connection under regime  $G$ , i.e.  $n_m = \sum_{j=1}^m r_j$  and let  $\chi_{n_m}$  be the contention at time slot  $\tau$  after the  $n_m$  slot requests have been revealed.<sup>12</sup> Suppose that  $k > 1$  is the first connection such that  $r_k > 1$  (there is at least one connection requesting one slot per period), then for  $m_1 = 1, \dots, k - 1$ ,  $\chi_{n_{m_1}} = \sum_{j=1}^{m_1} Z_j$ , where  $Z_1, Z_2, \dots, Z_{k-1}$  are independent Bernoulli random variables with success probability  $1/M$ , such that  $Z_i = 1$  represents whether connection  $i$  requests slot  $\tau$ .  $\chi_{n_{m_1}}$  are therefore  $\text{Bin}(n_{m_1}, 1/M)$  random variables:

$$\Pr\{\chi_{n_{m_1}} = \kappa\} = \binom{n_{m_1}}{\kappa} \left(\frac{1}{M}\right)^\kappa \left(1 - \frac{1}{M}\right)^{n_{m_1} - \kappa}. \quad (13)$$

We use  $m_2$  to index the connections requesting multiple slots per period starting with the  $k$ th one. For  $m_2 = 0, \dots, N - k$ , define  $Y_{m_2}$  as the number of requests for  $\tau$  from the  $k + m_2$ th connection. Since  $\tau$  can be chosen by at most one out of the  $r_{k+m_2}$  requests (each request covers a different segment of the period),  $Y_{m_2}, m_2 = 0, \dots, N - k$  are independent Bernoulli random variables with success probability  $1/(M/r_{k+i}) = r_{k+m}/M$ . Thus we have:

$$\begin{aligned} \chi_{n_{k+m_2}} &= \chi_{n_{k+m_2-1}} + Y_{m_2} \\ &= \chi_{n_{k-1}} + \sum_{j=0}^{m_2} Y_j, \end{aligned} \quad (14)$$

Note that (14) applies to the case where all connections request more than 1 slot per period ( $k = 1$ ) by adopting the convention  $\chi_{n_0} = 0$ .

In general, we can express the pmf of  $\chi_{n_j}$ , for  $j = 1, \dots, N$ , recursively as

$$\begin{aligned} \Pr\{\chi_{n_j} = \kappa\} &= \Pr\{\chi_{n_{j-1}} = \kappa\} \left(1 - \frac{r_j}{M}\right) \\ &\quad + \Pr\{\chi_{n_{j-1}} = \kappa - 1\} \left(\frac{r_j}{M}\right). \end{aligned} \quad (15)$$

<sup>11</sup>For simplicity, we consider only the case where  $M/r_j$  is an integer. The case where segments differ in size (by  $\pm 1$ ) is similar but tedious.

<sup>12</sup>Since all slots within the period are equally likely to be chosen by any given number of requests from any particular connection, we study contention at an arbitrary slot  $\tau$ . To simplify the notation, we use  $\chi_{n_m}$  in place of  $\chi_{\tau, n_m}$ .

Now, consider a regime  $G'$  identical to  $G$  at  $l$  (same set of connections and bandwidth allocations), except that the  $k$ th connection is exchanged with  $r_k$  connections each requesting one slot every period. The  $r_k$  requests are statistically independent: each request is for a slot chosen at random from within the  $M$  slots in the period. Thus, we have  $\chi'_{n_j} = \chi_{n_j}$ ,  $j = 1, \dots, k-1$ , and  $\chi'_{n_k} = \chi'_{n_{k-1}} + \sum_{i=k}^{k+r_k} Z_i$ , where as defined above,  $Z_i$  is a Bernoulli random variable with success probability  $1/M$  that assumes the value 1 iff connection  $i$  requests slot  $\tau$ .<sup>13</sup> It follows that  $G'$  has a configuration similar to  $G$ . Specifically, let  $n'_m$  and  $k'$  be defined under  $G'$  similarly to  $n_m$  and  $k$  under  $G$ . Then  $\chi'_{n'_m}$ ,  $m_1 = 1, \dots, k' - 1$  are binomial random variables defined as in (13), and (as in (14)) for  $m_2 = 0, \dots, |\mathcal{C}_i^{G'}| - k'$ ,  $\chi'_{n'_{k'+m_2}} = \chi'_{n'_{k'-1}} + \sum_{i=0}^{m_2} Y'_i$ , where  $Y'_i$  is a Bernoulli random variable that assumes the value 1 iff  $\tau$  is requested by the  $k' + i$ th connection under  $G'$ . Therefore,  $\Pr\{\chi'_{n_{k+j}} > \kappa\} \geq \Pr\{\chi_{n_{k+j}} > \kappa\}$  for  $j = 0, \dots, N - k$  and  $\kappa \geq 1$  implies that successively exchanging the remaining connections in  $G$  will similarly result in higher contention. In other words, to establish the necessary condition, it suffices to prove that  $\Pr\{\chi'_{n_{k+j}} > \kappa\} \geq \Pr\{\chi_{n_{k+j}} > \kappa\}$  for  $j = 0, \dots, N - k$  and  $\kappa \geq 1$ .

In the following we adopt the notation:  $P_n(\kappa) \triangleq \Pr\{\chi_n = \kappa\}$ ,  $P'_n(\kappa) \triangleq \Pr\{\chi'_n = \kappa\}$ ,  $Q_n(\kappa) \triangleq \Pr\{\chi_n > \kappa\}$ , and  $Q'_n(\kappa) \triangleq \Pr\{\chi'_n > \kappa\}$ , for any  $n$ . First we state some properties of contention pmfs under  $G$  and  $G'$ . From (13) and (14), we have

$$P_{n_{k+j}}(0) = \left(1 - \frac{1}{M}\right)^{n_k - r_k} \prod_{i=0}^j \left(1 - \frac{r_{k+i}}{M}\right),$$

and, similarly,

$$P'_{n_{k+j}}(0) = \left(1 - \frac{1}{M}\right)^{n_k} \prod_{i=1}^j \left(1 - \frac{r_{k+i}}{M}\right).$$

Also, from (15),

$$\begin{aligned} P_{n_{k+j}}(1) &= P_{n_{k-r_k}}(1) \prod_{i=0}^j \left(1 - \frac{r_{k+i}}{M}\right) \\ &\quad + P_{n_{k-r_k}}(0) \sum_{i=0}^j \frac{r_{k+i}}{M} \prod_{\substack{0 \leq m \leq j \\ m \neq i}} \left(1 - \frac{r_{k+m}}{M}\right), \end{aligned}$$

where,  $P_{n_{k-r_k}}(1) = \frac{n_k - r_k}{M} \left(1 - \frac{1}{M}\right)^{n_k - r_k - 1}$ .  $P'_{n_{k+j}}(1)$  has a similar expression where  $P'_{n_k}(1)$  replaces  $P_{n_{k-r_k}}(1)$  and the indexed sum and products run from 1 to  $j$ .

**Lemma 3** (*Properties of contention pmfs.*) For  $j = 0, \dots, N - k$

- (i)  $P'_{n_{k+j}}(0) > P_{n_{k+j}}(0)$
- (ii)  $Q'_{n_{k+j}}(1) > Q_{n_{k+j}}(1)$
- (iii)  $\exists \tilde{\kappa} > 1$  such that  $P'_{n_k}(\kappa) - P_{n_k}(\kappa) < 0$  for  $1 \leq \kappa < \tilde{\kappa}$  and  $P'_{n_k}(\kappa) - P_{n_k}(\kappa) \geq 0$  for all  $\kappa \geq \tilde{\kappa}$ .

**Proof:** We sketch the proof. Parts (i) and (ii) are straightforward using algebraic manipulation and the application of Bernoulli's inequality; noting that  $Q_{n_{k+j}}(1) = 1 - P_{n_{k+j}}(0) - P_{n_{k+j}}(1)$ ,

<sup>13</sup>Recall that  $n_j$  is the number of requests up to and including requests from the  $j$ th connection under regime  $G$ .

and similarly for  $Q'_{n_{k+j}}(1)$ . Parts (i) and (ii) together imply that  $P'_{n_k}(\kappa) - P_{n_k}(\kappa)$  changes sign at  $\kappa = 1$  (since (ii) implies  $P'_{n_k}(1) < P_{n_k}(1)$ ). The difference of the mass functions has to change sign at least at one higher value of  $\kappa$ . This is because under  $G'$ ,  $\tau$  can be chosen by more than one of the  $r_k$  requests. That is,  $P_{n_k}(n_k - r_k + i) = 0 < P'_{n_k}(n_k - r_k + i)$  for  $i = 2, 3, \dots, r_k$  (note that both pmfs are zero for all  $\kappa > n_k$ ). Part (iii) results by showing that for any  $\kappa > 1$ ,  $P'_{n_k}(\kappa) \geq P_{n_k}(\kappa)$  implies  $P'_{n_k}(\kappa + 1) \geq P_{n_k}(\kappa + 1)$ . ■

The following is a consequence of Lemma 3:

**Lemma 4**  $Q'_{n_k}(\kappa) \geq Q_{n_k}(\kappa)$  for all  $\kappa \geq 1$ .

**Proof:** Since  $P'_{n_k}(\kappa) - P_{n_k}(\kappa) < 0$  for  $1 \leq \kappa < \tilde{\kappa}$ ,  $Q'_{n_k}(\kappa) - Q_{n_k}(\kappa)$  is increasing in the interval  $[1, \tilde{\kappa})$ . Then, by Part (ii) of Lemma 3,  $Q'_{n_k}(\kappa) > Q_{n_k}(\kappa)$  for  $\kappa \in [1, \tilde{\kappa})$ . Furthermore, since  $P'_{n_k}(\kappa) - P_{n_k}(\kappa) \geq 0$  for all  $\kappa \geq \tilde{\kappa}$ , the desired inequality holds for  $\kappa \geq 1$ . ■

The lemma above states that contention at  $\tau$  is higher under  $G'$  after  $n_k$  requests have been scheduled. We now use induction to prove the necessary condition.

**Lemma 5**  $Q'_{n_{k+j}}(\kappa) \geq Q_{n_{k+j}}(\kappa)$  for  $0 \leq j \leq N - k$  and  $\kappa \geq 1$ .

**Proof:** The desired result has been proved for  $j = 0$  in Lemma 4. Suppose that  $Q'_{n_{k+j}}(\kappa) \geq Q_{n_{k+j}}(\kappa)$  for some  $j > 0$  and all  $\kappa \geq 1$ , and that, to reach contradiction,  $Q'_{n_{k+j+1}}(\kappa^*) < Q_{n_{k+j+1}}(\kappa^*)$  for some  $\kappa^* \geq 1$ . The case  $\kappa^* = 1$  contradicts Part (ii) of Lemma 3. For  $\kappa^* > 1$ , we get from (14)

$$Q_{n_{k+j}}(\kappa^*) + \frac{r_{k+j+1}}{M} P_{n_{k+j}}(\kappa^*) > Q'_{n_{k+j}}(\kappa^*) + \frac{r_{k+j+1}}{M} P'_{n_{k+j}}(\kappa^*),$$

or, equivalently,

$$\frac{r_{k+j+1}}{M} \left( P_{n_{k+j}}(\kappa^*) - P'_{n_{k+j}}(\kappa^*) \right) > Q'_{n_{k+j}}(\kappa^*) - Q_{n_{k+j}}(\kappa^*).$$

Since  $r_{k+j+1} < M$ , and by the induction hypothesis  $Q'_{n_{k+j}}(\kappa^*) - Q_{n_{k+j}}(\kappa^*) \geq 0$ , the inequality implies

$$P_{n_{k+j}}(\kappa^*) - P'_{n_{k+j}}(\kappa^*) > Q'_{n_{k+j}}(\kappa^*) - Q_{n_{k+j}}(\kappa^*).$$

By rearranging, we get

$$Q_{n_{k+j}}(\kappa^* - 1) > Q'_{n_{k+j}}(\kappa^* - 1),$$

which contradicts the induction hypothesis. Thus completing the proof. ■

### Sufficiency proof

Suppose that  $G \in \mathcal{G}$  satisfies the necessary condition, yet  $G$  is not a regime of maximal contention of  $\mathcal{G}$  at  $l$ . Then there is a regime  $G^* \in \mathcal{G}$  that satisfies the necessary condition and results in higher contention at  $l$  than  $G$ . Sufficiency is established by observing that regimes in  $\mathcal{G}$  satisfying the necessary conditions for maximal contention at  $l$  are statistically identical at that link: Under every regime in  $\mathcal{G}$  satisfying the necessary condition, each connection requests exactly one slot per link period, and all slots within the period are equally likely to be requested by any connection. Thus contradicting the claim that  $G$  not being a regime of maximal contention of  $\mathcal{G}$  at  $l$ .