Analysis and Design of Book-ahead Bandwidth-Sharing Mechanisms

Xiangfei Zhu, *Student Member, IEEE,* and Malathi Veeraraghavan, *Senior Member, IEEE*

Abstract

In this article, we present a novel discrete-time Markov chain model of book-ahead bandwidth-sharing mechanisms. We use this analytical model and a simulation model to understand the benefits of book-ahead (BA) bandwidth-sharing when compared to the immediate-request (IR) call-blocking mode of bandwidth-sharing in circuit-switched networks. We study two different BA schemes, `BA-all`, in which the caller accepts any set of available timeslots, and `BA-n`, in which the caller specifies n call-initiation time options. Numerical results show that the `BA-all` achieves 95% utilization with a call-blocking probability of only 1%, while in the IR mode, call blocking probability is 23% even when utilization is only 80%. The `BA-n` schemes perform as well as the `BA-all` scheme if the call-initiation time options are restricted to fall on timeslot boundaries separated by the minimum call holding time. The length of the advance-reservation horizon, K, is shown to increase linearly with call holding time, H. The ratio K/H is primarily dependent on the link capacity in channels. For example, if the link is divided into 10 channels, to achieve a 2% call blocking probability, the advance-reservation horizon needs to be a factor of 4 times the call holding time. In other words, the extra data storage and processing required to accept and maintain advance reservations is not significant.

Index Terms

Optical networking, Grid networks, GMPLS, advance reservations, circuit-switching.

I. Introduction

There is an increasing interest in optical circuit-/virtual circuit (VC)-switched networks to support the high-speed and predictable-service requirements of applications used in the scientific research community. For example, high-energy physics research experiments are expected to create terabyte-sized datasets, the sharing of which requires high-speed connectivity between geographically distributed organizations. Other scientific teams engaged in research fields, such as astrophysics, Fusion sciences, Earth sciences, and genomics, require network connectivity with predictable rate- and/or delay-guaranteed service for remote visualization of datasets, and remote control of computations and instruments.

The potential of optical networks to meet these high-speed and predictable-service requirements has led to the creation of various circuit-/VC-switched experimental testbeds [1]–[9]. The focus of networking research on these testbeds is on dynamic bandwidth sharing because of the large number of universities and national research laboratories involved in these scientific projects. Given that both scientists and resources (computation, storage, visualization, and instruments) are globally distributed, it becomes cost prohibitive to build dedicated networks for each scientific project. Instead, the high-capacity links of these network testbeds need be shared dynamically.

The need for dynamic bandwidth sharing has led to the adoption of Generalized Multi-Protocol Label Switched (GMPLS) control-plane protocols [10] in several of these testbeds. These control-plane protocols consist of signaling protocols, used to set up and release circuits on a call-by-call basis [11], and routing protocols, used to disseminate topology, reachability and loading conditions [12], necessary to support the call setup procedure.

The bandwidth-sharing mode implemented in the GMPLS signaling engines in switch controllers is an **immediate-request** call-blocking mechanism. As call requests arrive, bandwidth is allocated on a first-come first-served basis until the link capacity runs out at which point calls are blocked (rejected). This mode of bandwidth sharing is similar to that used in the circuit-switched telephony network.

In [13], we set out to understand whether this immediate-request mode of bandwidth sharing is sufficient for use in circuit-/VC-switched networks created for scientific applications. Specifically, we considered the high-speed requirement of these applications. Link capacity C can expressed as m channels, where the per-channel bandwidth is the crossconnect rate of the switches. For example, if a circuit switch has a crossconnect rate of 1Gb/s [14], then a 10Gb/s link is said to have 10 channels. The higher the per-circuit (channel) bandwidth, the lower the m.

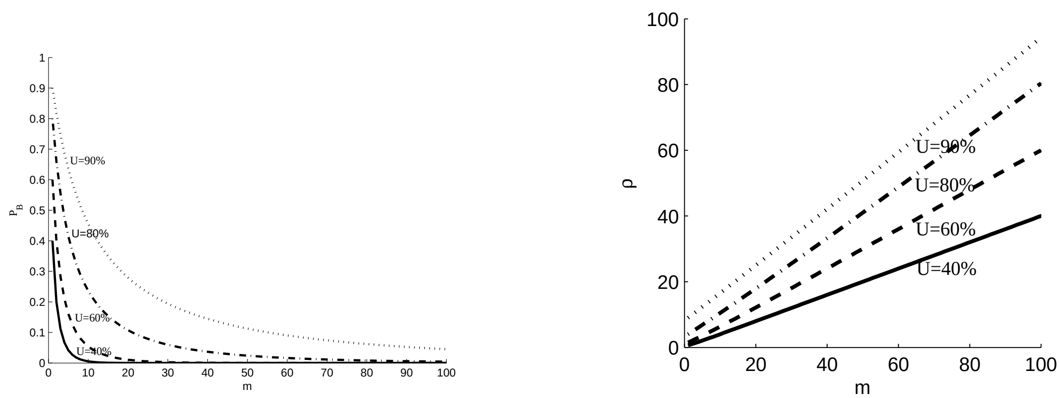
The immediate-request call-blocking mode of bandwidth sharing is well modeled with an

M/G/m/m system [15]. The Erlang-B formulas characterizing a single-link system compute:

$$P_B = \frac{\rho^m/m!}{\sum\limits_{i=0}^{m} \rho^i/i!}, \tag{1}$$

$$U = \frac{\rho(1 - P_B)}{m}, \tag{2}$$

where P_B denotes the call blocking probability, U denotes the system utilization, m is the link capacity in channels, and ρ, the offered traffic load, is given by $\rho = \lambda/\mu$, where λ is the call arrival rate and $1/\mu$ is the mean call holding time.



(a) P_B vs. m (b) ρ vs. m

Fig. 1: Plot of P_B and ρ vs. m for $U = 40\%$, 60%, 80%, and 90%.

Fig. 1 plots P_B vs. m and the corresponding traffic load vs. m for a few fixed values of utilization using (1) and (2). As can be seen in these plots, it is difficult to combine high utilization with a low call blocking probability when m is small. For example, when $m = 10$, to achieve 80% utilization, the corresponding call blocking probability will be as high as 23.62%. The system should be engineered to create the corresponding traffic load, ρ, for this point of operation.

From this simple analysis, it is apparent that if the per-circuit bandwidth requirement of applications is high relative to the shared link capacity, alternative bandwidth-sharing modes are required. We attempt a preliminary classification of bandwidth-sharing modes for circuit-/VC-switched networks in Fig. 2. This shows that if m is large, which means per-circuit bandwidth is low relative to link capacity, then the immediate-request mode is sufficient to achieve high utilization with low call blocking rates. But if m is small, we need to consider the mean call holding time. If the mean call holding time is "long," we reason that we would require a **book-ahead** bandwidth-sharing mechanism, i.e., one that allows users to reserve network bandwidth in advance. We expect call holding times to be long for some scientific applications. For example, terabyte-dataset transfers over Gb/s circuits will require multi-hour call holding times. Remote visualizations often require hour-long sessions, and remote access to instruments, as in the Fusion science project, could require day-long sessions.

If, on the other hand, m is small and the mean call holding time is also small, we speculate that a call queueing mode can be designed for bandwidth sharing. Since mean waiting time in a queueing system is proportional to mean call holding time, such a sharing mode is likely to be feasible only when the mean call holding time is small. We plan to design and evaluate call-queueing schemes as part of our future work.

The focus of this paper is on the book-ahead bandwidth sharing mechanism. The key question we answer in this paper is whether, by allowing advance reservations, a network operator can achieve both high utilization and low call-blocking probability. We present a **discrete-time Markov chain model (DTMC)** for **book-ahead (BA)** bandwidth sharing. We use this analytical model in combination with a simulation model to compare the BA mechanism with the IR mode, and to study the performance of different types of BA schemes. Our results show that with a BA mechanism lower call blocking probabilities can be achieved than with the immediate-request (**IR**) mode. For example, when $m = 10$, at a system utilization of 80%, the call blocking probability with the BA mechanism is about 2%, while with the IR mechanism it is 23.62%.

We compare the performance of two variants of the BA mechanism, `BA-n` schemes, in which the user specifies n call-initiation time options for the advance reservation, and `BA-all`, in which

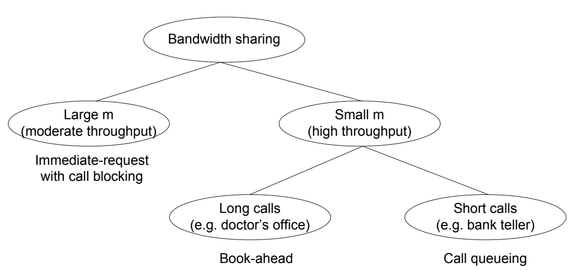


Fig. 2: A classification of bandwidth-sharing modes for circuit-/VC-switched networks.

the user accepts a reservation at any available time range. We allow for multiple call classes that differ in their call durations. While the `BA-all` scheme achieves the high-utilization, low-call-blocking-probability operating point under an unrestricted call-initiation time policy, it could be limiting to scientists who may want to specify a limited set of call-initiation time options based on their availability. With a restricted call-initiation time policy, in which users can only request starting times that are separated by the minimum call holding time, the `BA-n` schemes perform almost as well as the `BA-all` scheme. Finally, we show that the advance-reservation horizon, which is the maximum time up to which the scheduler accepts advance reservations, can be fairly small. It needs to be only 4 times the call holding time (in the single-class case) for a 2% call blocking probability when $m = 10$. This factor increases to 14 when m is only 2 channels.

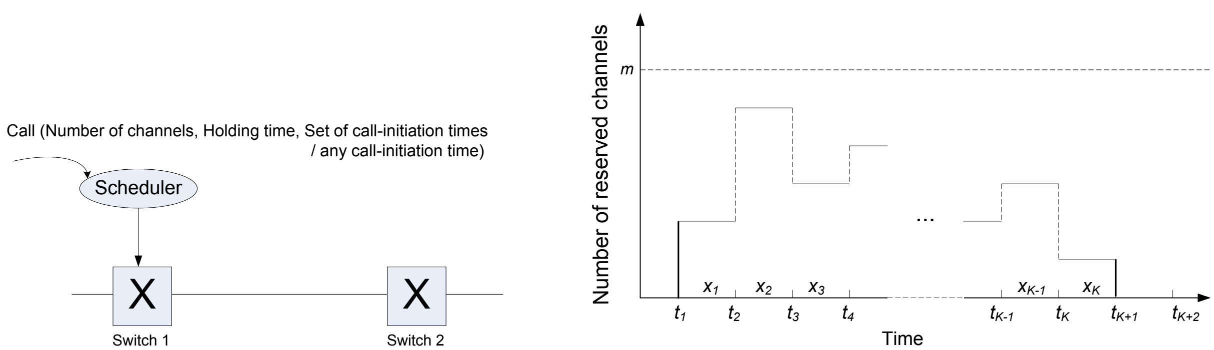
The remainder of this paper is structured as follows: Section II surveys related work on book-ahead mechanisms. Section III describes the book-ahead mechanism we investigate in this paper. In Section IV we describe our DTMC model of the book-ahead bandwidth-sharing system. Section V presents numerical results comparing our analytical and simulation models, comparing different bandwidth-sharing mechanisms (`BA-all`, `BA-n`, IR), and determining appropriate values for design parameters, such as the advance-reservation horizon. Section VI concludes the paper.

II. Related Work

There are several research papers on book-ahead bandwidth-sharing (also referred to as "advance reservation") schemes [16]–[22]. Wolf and Steinmetz [16] outlined a framework for a book-ahead mechanism and discussed design issues. Ferrari, Gupta, and Ventre [17] proposed a distributed book-ahead service. Naik, Siegel, and Chong [18] proposed a class and priority based mechanism to schedule calls in oversubscribed preemptive networks. Two types of calls were considered: i) calls requesting the file transfers, and ii) calls requesting a fixed amount of bandwidth for a fixed duration. A heuristic based on sorting, preemption, and repositioning was proposed. Yuan, Tham, and Ananda [19] investigated book-ahead sharing with flexible bandwidth allocation. To increase utilization and reduce blocking probabilities, they exploited the potential flexibility of calls using a probe-based adaptive reservation approach. Most of this papers use simulations to evaluate their book-ahead mechanisms.

Greenberg, Srikant, and Whitt [20] proposed a scheme for sharing resources among book-ahead calls and immediate-request calls to avoid utilization loss from strict bandwidth partitioning. They assume the arrival and service rates of immediate-request calls are much higher than those of book-ahead calls, and obtain approximate estimates of system performance by decomposing a 2-D Markov chain model and analyzing book-ahead and immediate-request calls separately. Coffman, Jelenkovic, and Poonen [21] mathematically analyzed the fraction of time that a resource is booked in a book-ahead system with a Poisson call arrival process. Virtamo [22] derived a closed-form solution for a single-server book-ahead system with deterministic call holding times of one or two timeslots, and an asymptotic analytical solution for a general holding time distribution. Numerical results for multi-server systems were provided by simulations. None of the above mentioned work considers book-ahead calls with multiple acceptable options. Interestingly, our results show that a book-ahead mechanism that only specifies one call-initiation time may perform worse than an immediate-request mechanism when the system load is high.

III. BOOK-AHEAD MECHANISM



(a) Book-ahead mechanism model

(b) Bandwidth reservation status maintained by the scheduler

Fig. 3: Book-ahead mechanism.

Our book-ahead mechanism for sharing the capacity of a single link works as follows. The link capacity C is discretized and expressed as m channels where the bandwidth of each channel is C/m. Time is also discretized for simpler management. Fig. 3a illustrates our system model with the scheduler managing the bandwidth of the link between switch 1 and switch 2. The scheduler maintains a database of reserved bandwidth as a function of time as illustrated in Fig. 3b. The maximum time horizon up to which the scheduler will accept reservation requests, as shown in Fig. 3b, is K. A call arriving at time t_1 can request one or more channels for any contiguous set of timeslots between t_2 and t_{K+2}.

As shown in Fig. 3a, a user requests bandwidth as a discrete number of channels and specifies call holding time as a discrete number of timeslots. A user also provides a preference-ordered acceptable set of call-initiation times (up to n options), or sets the indicator that any call-initiation time is acceptable. Calls accepting any initiation time are referred to as `BA-all` calls in this paper. Calls with specific n acceptable initiation times are referred to as `BA-n` calls. A `BA-all` call is scheduled as early as possible, while in a `BA-n` call the first admissible option is scheduled from the user's preference-ordered list. A `BA-n` call arriving at time t_1, requesting a channel for H timeslots, will be blocked if a channel is not available for H timeslots starting from any of the n options. A `BA-all` call will be blocked if a channel is not available for H contiguous timeslots in the entire K-timeslot horizon.

IV. Analytical Model

We use a Discrete-time Markov Chain (DTMC) to model the book-ahead mechanism. We state our assumptions, derive the transition probability matrix, and provide an example.

A. Assumptions

We make the following assumptions:

1. Calls arrive according to a Poisson process with an aggregate rate λ. Measurements have shown that user-initiated TCP session arrivals on the Internet can be modeled as Poisson processes [23], and telephony traffic has long been modeled as a Poisson process [15]. Assuming that the new applications for which the book-ahead bandwidth sharing mechanism is being designed will follow similar trends, we make the Poisson process assumption for call arrivals.
2. Further, we approximate the exponential distribution of call interarrival times with a geometric distribution because time is discretized in our book-ahead mechanism (see Section III). The smallest value of call interarrival time is thus one timeslot. The parameter of the geometric distribution, p, represents the probability that a call arrives within a timeslot. In other words, in each timeslot either only one call arrives or no call arrives (which happens if the call interarrival time is multiple timeslots in length). Conditions under which this

approximation is valid will be discussed in a later section that describes our selection of numerical values for the parameters of this model. We will see that the timeslot length needs to be very small.

3. There are l classes of calls. The arrival rate of class-i calls is λr_i, $1 \leq i \leq l$. Fig. 3a shows that each call request specifies the required number of channels, the call holding time, and its preference regarding call-initiation times. In our model, we assume that calls of all classes request only one channel but differ in their holding time requests. The holding time of a class-i call is h_i timeslots, $1 \leq i \leq l$.
4. In the BA−n scheme, the n call-initiation time options, denoted by $s_1, s_2, ..., s_n$, are uniformly distributed among the $(K - h_i + 1)$ eligible timeslots for a class-i call. The n options are assumed to be different from each other.
5. The time for reservation processing by the scheduler is considered negligible in the model, which allows us to have the system change states instantaneously on timeslot boundaries as calls arrive and depart.

B. State transition probability matrix

We denote the state of the system by vector $\boldsymbol{x}$, which consists of K components $(x_1, x_2, ..., x_K)$, where x_i represents the number of channels that have been reserved for the i^{th} timeslot. For example, x_1 indicates the number of channels that have been reserved for the current timeslot (the first timeslot). The state space S is therefore defined as

$$S \triangleq \{\boldsymbol{x} = (x_1, x_2, ..., x_K) : 0 \leq x_i \leq m \text{ for } i = 1, 2, ..., K\}.$$

The size of the state space

$$N = (m + 1)^K. \tag{3}$$

Calls arrive on timeslot boundaries and since we neglect processing time, state transitions occur instantaneously at each timeslot boundary. If, at time t_{1^+} (just after time instant t_1) the system state $S_{t_{1^+}}$ is $(x_1, x_2, ..., x_K)$ as shown in Fig. 3b, the system will be in the same state at t_{2^-} (just before time instant t_2). If no call arrives at time instant t_2 or the arriving call is blocked, the system will transition to state $(x_2, x_3, ..., x_K, 0)$ at time t_2. This is because x_2, which is the number of channels reserved for the (t_2, t_3) timeslot when the system is in state S_{t_1}, becomes the first component of the state vector at time t_2. Similarly, x_3, the number of channels reserved for the (t_3, t_4) timeslot, which was the third component of the state vector S_{t_1}, becomes the second component of the system state S_{t_2}. In other words, all components of the state vector shift to the left by one at each timeslot boundary.

If a call arrives at time instant t_2 and successfully reserves some timeslots, the system state will change accordingly in addition to the above described left-shift of the components in the state vector at t_2. For example, if the system is in state $(0, 1, 0)$ at time t_{1+} and a call arrives at t_2 and reserves a channel with a call-initiation time of t_3 and a holding time of two timeslots, then the system will transition to state $(1, 1, 1)$ at time t_2. Based on these observations, we can see that the state transition probabilities from one state to another only depend on the current state and are independent of the system history. Therefore this model is a discrete-time Markov chain. The model is also time-homogeneous because the call arrival and departure processes are assumed to be stationary.

Before we derive the transition matrix of this Markov chain, for ease of description, we define a left shift operator "$\leftarrow$"on vector $\boldsymbol{x}$ as follows. If $\boldsymbol{x} = (x_1, x_2, ..., x_K)$, $\overleftarrow{\boldsymbol{x}} = (x_2, x_3, ..., x_K, 0)$. We also define a K-component vector $\boldsymbol{O}_{i,j} = (O_1, O_2, ..., O_K)$, where

$$O_x \triangleq \begin{cases} 1 & \text{if } i \leq x \leq j, \\ 0 & \text{otherwise.} \end{cases}$$

The transition probability from state $\boldsymbol{x}$ to state $\boldsymbol{y}$, denoted by $p_{\boldsymbol{xy}}$, is

$$p_{\boldsymbol{xy}} = \begin{cases} 1 - p + pB_{\boldsymbol{x}} & \text{if } \boldsymbol{y} = \overleftarrow{\boldsymbol{x}}, \\ pr_j q_{i,j} & \text{if } \boldsymbol{y} = \overleftarrow{\boldsymbol{x}} + \boldsymbol{O}_{i-1, i+h_j-2}, \text{ and } \boldsymbol{y} \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where p denotes the probability that a call arrives in a timeslot, r_j denotes the probability that the arriving call belongs to class-j, $q_{i,j}$ denotes the probability that a class-j call is admitted with a call-initiation time of the i^{th} timeslot relative to its arrival instant, and $B_{\boldsymbol{x}}$ denotes the probability that an incoming call is blocked when the system is in state $\boldsymbol{x}$.

The first row of (4) represents the case when no call arrives (probability of $(1 - p)$) and the case when a call arrives (probability of p) but is blocked (probability of $B_{\boldsymbol{x}}$). Therefore the state vector only shifts to the left with no new reservations. The second row represents the case that the call is admitted with a call-initiation time of the i^{th} timeslot away from the call arrival instant. Since the scheduler is able to reserve a channel for h_j consecutive timeslots starting at the i^{th} timeslot relative to the call arrival instant, the components $x_i, x_{i+1}, ...,$ and x_{i+h_j-1} of the state vector should be smaller than the link capacity, m. The last row shows that the transition probabilities to all other states are 0.

To calculate $q_{i,j}$, recalling that any of the n call-initiation time options, $s_1, s_2, \ldots, s_n$, could be the one that requested the i^{th} timeslot, we first determine the probability that the t^{th} option was the one that requested the i^{th} timeslot, i.e., $s_t = i$. Before we can compute this probability, we define d_j to be the number of timeslots that are "ineligible," i.e., that the starting instants of these timeslots, if specified as one of the call-initiation options, would lead to a blocked call because of a lack of resources in one or more of the following h_j timeslots. We can determine d_j for any given state $\boldsymbol{x}$ by examining the state vector and finding all those elements for which at least one or more of the following h_j timeslots are fully booked.

Successfully admitting the call with its channel reservation starting from the i^{th} timeslot implies that the i^{th} timeslot does not belong to one of the ineligible d_j timeslots. Furthermore, if this timeslot was requested as the t^{th} call-initiation time option, i.e., $s_t = i$, it means that the previous $(t-1)$ options were all denied. The implication of this statement is that the first $(t-1)$ options are all ineligible, i.e., they all belong to the set of d_j ineligible timeslots. Finally since we require the n call-initiation time options to be distinct, while the first option could be any one of the d_j ineligible timeslots, the second option should necessarily be selected from one of the remaining $(d_j - 1)$ ineligible timeslots. We can apply reasoning similar to that needed in determining the hypergeometric probability mass function, $h(k; n, d, N)$, which is the probability of having k defective units in a random batch of n elements, drawn *without replacement* from a larger set of N elements, which is known to contain d defective elements [24]. This probability is:

$$h(k; n, d, N) = \frac{\binom{d}{k}\binom{N-d}{n-k}}{\binom{N}{n}}, \text{ for } k = 0, 1, ..., \min\{d, n\},\ n - k \leq N - d. \tag{5}$$

Our current problem is similar in the sense that once a timeslot is chosen for a particular call-initiation time option, it cannot be selected for a subsequent option. In effect, it is selection *without replacement*. There are totally $(K + 1 - h_j)$ candidate timeslots (analogous to elements in the hypergeometric definition), which is known to contain d_j ineligible timeslots (defective elements). Using similar reasoning, we determine the probability $R_{t,i,j}$ that the first acceptable call-initiation time in a class-j call is its t^{th} option and that $s_t = i$ as:

$$R_{t,i,j} = P(s_1, s_2, ..., s_{t-1} \text{ are ineligible}) \cdot P(s_t \text{ is eligible} \mid s_1, s_2, ..., s_{t-1} \text{ are ineligible}) \cdot P(s_t = i \mid s_1, s_2, ..., s_{t-1} \text{ are ineligible and } s_t \text{ is eligible}). \quad (6)$$

The first factor is analogous to the probability that all elements in a random batch of $(t-1)$ elements are defective. Therefore, by applying (5) we have

$$P(s_1, s_2, ..., s_{t-1} \text{ are ineligible}) = \begin{cases} h(t-1; t-1, d_j, K+1-h_j) = \dfrac{\binom{d_j}{t-1}\binom{K+1-h_j-(t-1)}{t-1-(t-1)}}{\binom{K+1-h_j}{t-1}} & t-1 \leq d_j, \\ 0 & \text{otherwise.} \end{cases}$$

Reducing the above,

$$P(s_1, s_2, ..., s_{t-1} \text{ are ineligible}) = \begin{cases} \prod_{y=0}^{t-2} \frac{d_j - y}{K+1-h_j-y} & t-1 \leq d_j, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Consider the second factor. For the t^{th} option, given the *without-replacement* selection procedure, only $(K+1-h_j-(t-1))$ timeslots remain but the number of eligible timeslots are still $(K-h_j-d_j+1)$ because all timeslots selected in the first $(t-1)$ options are ineligible timeslots. Therefore:

$$P(s_t \text{ is eligible} \mid s_1, s_2, ..., s_{t-1} \text{ are ineligible}) = \frac{\binom{K-h_j-d_j+1}{1}}{\binom{K+1-h_j-(t-1)}{1}} = \frac{K-h_j-d_j+1}{K-h_j-t+2}. \quad (8)$$

Similarly we can obtain the third factor:

$$P(s_t = i \mid s_1, s_2, ..., s_{t-1} \text{ are ineligible and } s_t \text{ is eligible}) = \frac{1}{K-h_j-d_j+1}. \quad (9)$$

Combining (7), (8) and (9) with (6), we get:

$$R_{t,i,j} = \begin{cases} \frac{1}{K-h_j-t+2} \prod_{y=0}^{t-2} \frac{d_j - y}{K+1-h_j-y} & t-1 \leq d_j, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Since any of the n call-initiation time options could be the selected t^{th} option and $q_{i,j}$ is the probability that a class-j call is admitted with a call-initiation time of the i^{th} timeslot relative to its arrival instant,

$$q_{i,j} = \begin{cases} \sum_{t=1}^{n} R_{t,i,j} & \text{if the } i^{th} \text{ timeslot is an eligible timeslot,} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Having computed $q_{i,j}$, we now have the state transition probabilities for the second row of (4). We next turn our attention to the blocking probability $B_{\boldsymbol{x}}$ in state $\boldsymbol{x}$ in order to compute the state transition probabilities corresponding to the first row of (4). Before we calculate $B_{\boldsymbol{x}}$, we compute the blocking probability for a class-j call, denoted $B_{\boldsymbol{x},j}$. Recalling that a call is blocked if and only if resources are unavailable at all the n options specified by the user, $B_{\boldsymbol{x},j}$ is analogous to the probability that all elements in a random batch of n elements are defective in the hypergeometric example. Therefore $B_{\boldsymbol{x},j}$ can be calculated by applying (5) as follows:

$$B_{\boldsymbol{x},j} = \begin{cases} h(n; n, d_j, K+1-h_j) = \prod_{y=0}^{n-1} \frac{d_j - y}{K+1-h_j-y} & \text{if } n \leq d_j, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We find the aggregate call blocking probability $B_{\boldsymbol{x}}$ using a weighted average

$$B_{\boldsymbol{x}} = \sum_{j=1}^{l} (r_j B_{\boldsymbol{x},j}), \tag{13}$$

where the weight r_j is the probability that the arriving call is a class-j call. Having computed $q_{i,j}$ and $B_{\boldsymbol{x}}$, we now have the DTMC transition probability matrix specified in (4).

C. Example

We derive the transition probability matrix for the following example. Let m, the link capacity in channels, be 1, and K, the advance-reservation horizon, be 3 timeslots. We allow two call classes ($l = 2$). The holding time for class-1 calls (h_1) is 1 and for class-2 calls (h_2) is 2. The number of call-initiation time options allowed, n, is 1. Fig. 4 shows the transition diagram and Table I shows the transition probability matrix for this example. We explain in detail how we apply (4)-(13) to determine the transition probabilities from state (0, 0, 1).

Assume the system is in state (0,0,1) at time t_1. If no call arrives at time t_1 or a call arrives but

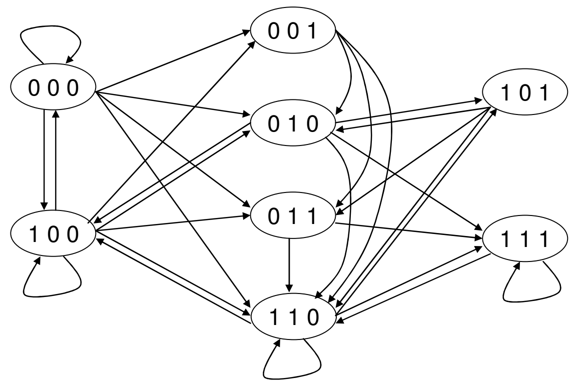


Fig. 4: DTMC for an example system: $m = 1$, $K = 3$, $l = 2$, $h_1 = 1$, $h_2 = 2$, and $n = 1$.

gets blocked, the system will transition to state (0,1,0). To calculate $p_{(0,0,1),(0,1,0)}$, the transition probability from state (0,0,1) to state (0,1,0), we first find the number of ineligible timeslots in state (0,1,0) for the two call classes, $d_1 = 1$ and $d_2 = 2$. Using $K = 3$, $n = 1$, $l = 2$, $h_1 = 1$, $h_2 = 2$ in (12) we obtain

$$B_{(0,0,1),1} = \frac{d_1 - 0}{K + 1 - h_1 - 0} = \frac{1}{3}$$

and

$$B_{(0,0,1),2} = \frac{d_2 - 0}{K + 1 - h_2 - 0} = 1.$$

TABLE I: Transition matrix for the example Markov chain shown in Fig. 4.

	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
(0, 0, 0)	$1-p$	$\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$	$\frac{1}{2}pr_2$	$\frac{1}{3}pr_1$	0	$\frac{1}{2}pr_2$	0
(0, 0, 1)	0	0	$1-\frac{2}{3}pr_1$	$\frac{1}{3}pr_1$	0	0	$\frac{1}{3}pr_1$	0
(0, 1, 0)	0	0	0	0	$1-\frac{2}{3}pr_1-\frac{1}{2}pr_2$	$\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$	$\frac{1}{2}pr_2$
(0, 1, 1)	0	0	0	0	0	0	$1-\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$
(1, 0, 0)	$1-p$	$\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$	$\frac{1}{2}pr_2$	$\frac{1}{3}pr_1$	0	$\frac{1}{2}pr_2$	0
(1, 0, 1)	0	0	$1-\frac{2}{3}pr_1$	$\frac{1}{3}pr_1$	0	0	$\frac{1}{3}pr_1$	0
(1, 1, 0)	0	0	0	0	$1-\frac{2}{3}pr_1-\frac{1}{2}pr_2$	$\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$	$\frac{1}{2}pr_2$
(1, 1, 1)	0	0	0	0	0	0	$1-\frac{1}{3}pr_1$	$\frac{1}{3}pr_1$

Applying (13) we get

$$B_{(0,0,1)} = r_1 B_{(0,0,1),1} + r_2 B_{(0,0,1),2} = \frac{1}{3} r_1 + r_2 = 1 - \frac{2}{3} r_1.$$

Now we compute the probability $p_{(0,0,1),(0,1,0)}$ by applying the first row of (4)

$$p_{(0,0,1),(0,1,0)} = 1 - p + pB_{(0,0,1)} = 1 - p + p(1 - \frac{2}{3} r_1) = 1 - \frac{2}{3} p r_1.$$

If a class-1 call arrives at time t_1 and requests a call-initiation time of t_2, the system will transition to state (1, 1, 0). To calculate the probability of this transition, we first calculate $q_{2,1}$ by applying (11)

$$q_{2,1} = R_{1,2,1} = \frac{1}{K - h_1 - t + 2} = \frac{1}{3 - 1 - 1 + 2} = \frac{1}{3}.$$

Then we can calculate $p_{(0,0,1),(1,1,0)}$ by applying the second row of (4)

$$p_{(0,0,1),(1,1,0)} = p r_1 q_{2,1} = \frac{1}{3} p r_1.$$

If a class-1 call requests a call-initiation time of t_4, it will transition to (0,1,1) for which we can similarly compute the transition probability to be $\frac{1}{3} p r_1$. If the arriving call is a class-2 call which in this example means it requires two contiguous timeslots ($h_2 = 2$), all three timeslots are ineligible. This is because the channel is reserved for the (t_3 t_4) timeslot. If the call-initiation time requested is t_2, then only one timeslot (t_2 t_3) is available. If the requested call-initiation time is t_3, then again the call is blocked because timeslot (t_3 t_4) is reserved in state (0,0,1). Finally a class-2 call cannot request t_4 since the advance-reservation horizon K is three, which does not permit the required two timeslots if t_4 is the call-initiation time.

The remaining entries in the state transition matrix for this example, shown in Table I, can be derived in a similar manner. As can be observed from the table, the transition probabilities are independent of the reservation status in the first timeslot (the first component of vector $\boldsymbol{x}$) because the information in this timeslot is irrelevant when the state shifts to the left by one at the end of the timeslot. Thus rows in the table that differ only in the first component, e.g., (0,0,1) and (1,0,1), have identical entries in all the columns.

D. Performance metrics

To characterize the performance of the book-ahead sharing mechanism, we use the following two metrics: P_B, the long-run blocking probability for calls, and U, the long-run utilization of the system.

To calculate these two metrics, we first calculate the steady-state probabilities, denoted by vector $\boldsymbol{\pi}$, of this discrete-time Markov chain using the well-known formula [15]

$$\boldsymbol{\pi} = \boldsymbol{e}(\boldsymbol{I} + \boldsymbol{E} - \boldsymbol{P})^{-1} \tag{14}$$

where $\boldsymbol{P}$ denotes the transition matrix, $\boldsymbol{I}$ denotes the $(N \times N)$ matrix having ones along the main diagonal and zeros elsewhere, and $\boldsymbol{E}$ and $\boldsymbol{e}$ are respectively the $(N \times N)$ matrix and the $(1 \times N)$ row in which all elements equal one. After obtaining the steady-state probability vector $\boldsymbol{\pi}$, we can calculate the call blocking probability and system utilization.

1) Call blocking Probability: We denote the call blocking probability in each state of the model by vector $\boldsymbol{B}$, where elements $\boldsymbol{B_x}$ are calculated using (13). The long-run average blocking probability P_B can then be calculated as

$$P_B = \boldsymbol{\pi} \boldsymbol{B}^T. \tag{15}$$

2) Utilization: Similarly, we denote the utilization of the system in state $\boldsymbol{x} = (x_1, x_2, ..., x_K)$ by $\boldsymbol{u_x}$, which can be calculated as

$$\boldsymbol{u_x} = \frac{\sum_{i=1}^{K} x_i}{K \cdot m}. \tag{16}$$

Then the average utilization U can be calculated by

$$U = \boldsymbol{\pi} \boldsymbol{u}^T, \tag{17}$$

where the components of the vector $\boldsymbol{u}$ are the utilization computed in each state $\boldsymbol{u_x}$.

V. Numerical Results

We start with a discussion on how we select numerical values for significant input parameters. Among these are the call arrival probability (p), the link capacity in channels (m), average call holding time ($E[H]$), and the advance-reservation horizon (K). **Section V-A** describes how we select numerical values for these parameters.

Due to the state-space explosion problem (see (3)), the DTMC can only be solved for small values of m. Hence we implement a simulation model, and compare numerical results obtained from the two models for the $m = 1$ case. The simulation model is an event-driven system written in C++. Simulations for each variant of the book-ahead (BA) mechanism and the immediate-request (IR) mechanism are run for sufficiently long durations and repeated multiple times to obtain a large number of samples. As noted in [25], using alternative representations of the BA mechanism, i.e., our DTMC analytical model and a simulation model, we were able to uncover modeling errors and incorrect implementations of assumptions. Having obtained a close match of results from both models for the $m = 1$ case, we now have confidence in the models. **Section V-B** shows these results for the $m = 1$ case.

For larger values of m, we obtain numerical results using the simulation model. **Section V-C** presents these results. We simulated the system for different traffic loads consisting of a single call class as well as multiple call classes. Our principal finding was that while the `BA-all` bandwidth-sharing mechanism lowers call blocking probability significantly relative to the IR scheme for small m, the performance gains obtained with the `BA-n` schemes decreases as load increases. The reason for this decrease is as follows. Since the n call-initiation time options specified in a `BA-n` scheme are unrestricted and assumed to be uniformly distributed across the whole advance-reservation horizon, the requested reservation periods for the call could overlap with the multi-timeslot ranges of already-accepted reservations, in which case, the call is blocked.

In **Section V-D**, we restrict call-initiation time options to timeslot boundaries separated by the minimum call holding time (amongst the l call classes). Further, we assume that the call holding time of each class is restricted to be an integer multiple of the minimum call holding time. Under this restricted call-initiation times policy, we found that even `BA-n` schemes achieve the expected lowering of call blocking probability for small m.

A. Selection of numerical values for the model parameters

In section IV we stated our assumption that call interarrival times are exponentially distributed. Further we noted that we use a discrete random variable with the geometric distribution to approximate call interarrival times. As noted in [26], if a random variable X is exponentially distributed with parameter λ, then $Y = \lceil X \rceil$ is a geometric random variable with **parameter** $\boldsymbol{p} = 1 - e^{-\lambda}$. Approximating X by $\lceil X \rceil$ is clearly inaccurate if the probability that $X < 1$ is very large. For example, if λ is 2 calls/second, then $P(X \leq 1) = 0.8647$. If $X \leq 1$, Y is set to 1, which makes the approximation inaccurate.

Consider the mean values of these two random variables, $1/\lambda$ and $1/p$; we can theoretically calculate the maximum value of λ for any given desired maximum deviation of the corresponding p from λ. If we want the deviation to be less than a given value σ, which means

$$\frac{|p - \lambda|}{\lambda} = |\frac{p}{\lambda} - 1| = |\frac{1 - e^{-\lambda}}{\lambda} - 1| < \sigma, \tag{18}$$

we can calculate the maximum value of λ that satisfies the above inequality for different values of σ. For example, if we want the deviation to be less than 5%, λ should be less than 0.104. If we want σ to be less than 2%, λ should be less than 0.041.

Any λ can be downscaled to a smaller value that meets the above requirement by changing the time unit, e.g., 100 calls/sec can be represented as 0.1 calls/ms. By downscaling we essentially divide time into small enough timeslots so that the geometric distribution is a good approximation of the exponential distribution.

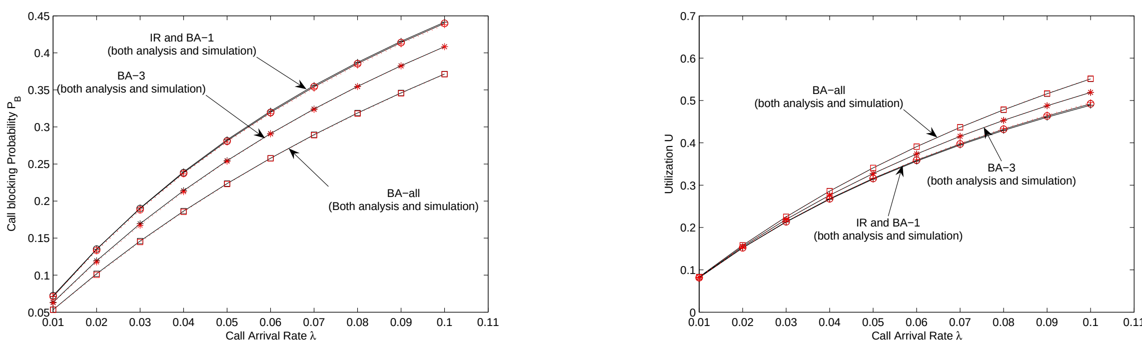
The next parameter we consider is m, the **link capacity** in channels. The number of the states in the discrete-time Markov chain increases exponentially with m (see (3)). The size of the transition matrix quickly exceeds the maximum matrix size that our programming environment allows. Therefore, we limit m to 1 for the numerical results generated from our DTMC model. For simulations, we choose values of 2, 5, and 10 for m. As described in Section I, scientific applications require large per-circuit bandwidth, which results in this range of m.

When m increases (say from 1 to 10), the **holding time** $\boldsymbol{H}$ also needs to be increased to study the system under a high load because the system load $\rho/m = \frac{\lambda E[H]}{m} = \frac{pE[H]}{m}$, where $E[H]$ is the mean call holding time. For example, if $m = 2$, in order to create a high load ($\rho/m = 0.99$) when $p = 0.09$, $\max_{j=1,2,\ldots l}(h_j)$ must be at least 22. Therefore, for the analytical results (when $m = 1$) we chose holding times on the order of 10, but increase the values to be on the order of 100 when $m = 10$.

The final parameter, the **advance-reservation horizon** K, depends upon the largest call holding time and the number of call-initiation time options, n. In the above example, when

$\max_{j=1,2,\ldots l}(h_j) = 22$, K must be at least 24 if $n = 3$ since call-initiation options should be distinct. The number of states in this example will be $(2+1)^{24}$ when $m = 2$ (see (3)), which makes the DTMC model difficult to solve on a 32-bit computer. For the analytical results, we choose a maximum call holding time of 9 and n values of 1 and 3, which makes it sufficient to choose a K value of 11. For simulation results when holding time is on the order of 100 timeslots, we choose K to be on the order of 1000. This is because under the restricted call-initiation time policy, K needs to be $(n+1)$ times $\max_{j=1,2,\ldots l}(h_j)$.

B. Comparison of analytical and simulation results



(a) Call blocking probability

(b) Utilization

Fig. 5: Performance of different bandwidth sharing mechanisms. $m = 1$, $l = 2$, $h_1 = 7$, $h_2 = 9$, $r_1 = 0.1$, $r_2 = 0.9$, and $K = 11$.

In this subsection, we present and compare results from the $m = 1$ case. We choose two classes of calls. The holding time of class-1 calls, h_1, is 7 timeslots and the arrival rate factor $r_1 = 0.1$, while the holding time of class-2 calls, h_2, is 9 timeslots and the arrival rate factor $r_2 = 0.9$. The advance-reservation horizon K is assumed to be 11 timeslots. Fig. 5 plots the results from four bandwidth sharing mechanisms: immediate-request (IR), book-ahead with 1 option (`BA-1`), book-ahead with 3 options (`BA-3`), and book-ahead that accepts any timeslot (`BA-all`). The plots show an excellent match of results from the analytical and simulation models.

Also, Fig. 5 shows that `BA-1` results in a call blocking probability that is similar to the IR scheme, while `BA-3` and `BA-all` result in lower call blocking probabilities and higher utilizations. Utilization, as seen in the Erlang-B formula (2), will reach a maximum value of 0.495 since m is only 1. The BA schemes improve this value marginally.

The performance gains of the BA schemes relative to the IR schemes for this set of input parameters is not significant. This is because m is only 1 here and K is small relative to h_2. The performance gains will be more clear in the next section when we increase m, and correspondingly increase K.

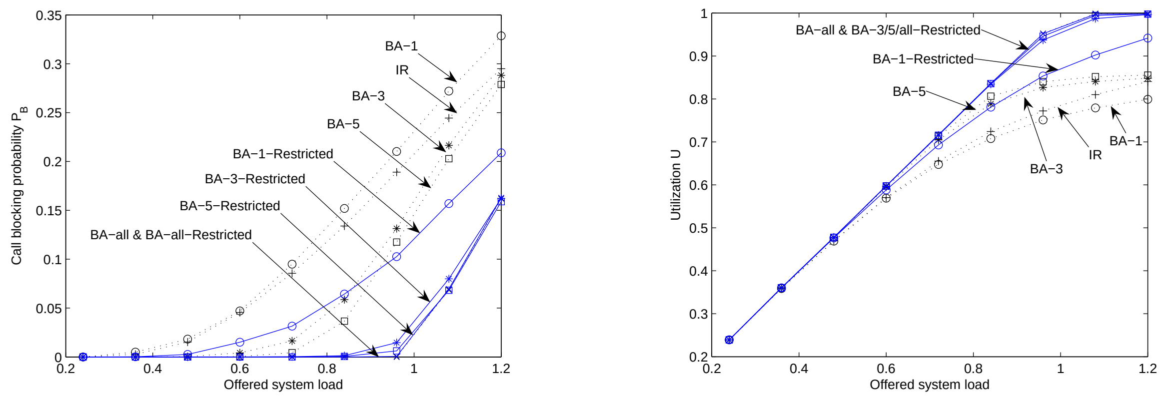
C. Simulation results for larger values of m

1) Single call class: In this subsection, we assume all calls belong to one class, which means $l = 1$. We fix the call holding time but vary call arrival rates to study the performance of different bandwidth-sharing mechanisms under increasing load. The link capacity, m, is set to 10 channels. First, we set call holding time H (we use H instead of h_1 to denote the call holding time of class-1 calls because there is only one call class) to 300 timeslots and change the call arrival probability p from 0.008 to 0.04. As noted in section V-A, the offered system load can be computed by $\frac{pE[H]}{m}$. Therefore this call arrival rate corresponds to an offered system load of 24% to 120%. The offered load can be higher than 100% because calls can be blocked. The advance-reservation horizon K is fixed at 2000 timeslots.

The call blocking probability and system utilization under different sharing mechanisms are plotted with dashed lines in Fig. 6. We see that the book-ahead mechanism in which calls accept any available timeslot (`BA-all`) performs much better than the immediate-request mechanism. A 95% utilization is achievable at a call blocking probability of 2% when the `BA-all` scheme is used, while that high level of utilization is not even achievable with the IR scheme. A 100% utilization with `BA-all` is achievable, at which point, the call blocking probability is 6%.

The `BA-n` mechanisms plotted with dashed lines in Fig. 6, `BA-3` and `BA-5`, also clearly outperform IR when the system load is high but not overloaded (between 50% and 100%). In other words, we can achieve high utilization while keeping call blocking probability low even when applications require a high per-channel capacity relative to link capacity.

However, if only one call-initiation time option (`BA-1`) is provided, then the call blocking probability and utilization are worse than with the IR mechanism. This is caused by our modeling assumption about call-initiation times. Since we assume that call-initiation times are uniformly selected from amongst $(K-h_j+1)$ timeslots, there is a high probability that the reservation period



(a) Call blocking probability

(b) Utilization

Fig. 6: Performance of different bandwidth sharing mechanisms. $m = 10$, $l = 1$, $H = 300$, and $K = 2000$.

requested, starting from the single call-initiation time option specified, overlaps with the holding time (which is multiple timeslots, specifically 300 in Fig. 6) of an already-admitted call. This probability increases with system load. This is also the reason why the performance gains of the BA-n schemes ($n > 1$) relative to the IR scheme decrease as offered load increases above 95%. While our assumption with regards to call-initiation time options was required for the DTMC model, these results demonstrate the need to limit users to certain call-initiation time options based on call holding times. Without such constraints, "gaps" could form in the reservation timeline. A "gap" is a time period smaller than the call holding time in which the system is not fully utilized. Gaps are caused by book-ahead calls requesting arbitrary call-initiation times, independent of the call holding time. For example, Fig. 7 shows a gap in the reservation status of a 3-channel single-class system, in which the call holding time, H, is 4 timeslots. In this particular instance, the scheduler has accepted call-initiation time requests for two new calls at t_6, while the last instant at which the system was fully reserved was the end of timeslot (t_2, t_3). Thus, in the three-timeslot range, (t_3, t_4), (t_4, t_5), and (t_5, t_6), the link occupancy is less than 3 channels. This becomes a "gap" since it is not long enough to admit a new 4-timeslot call. We consider a restricted call-initiation time policy, which would prevent such advance reservations, in the next section.

To study the impact of call holding time, we changed the call holding time H to 100 timeslots. We also changed the call arrival probability to range between 0.024 and 0.12 so that the same

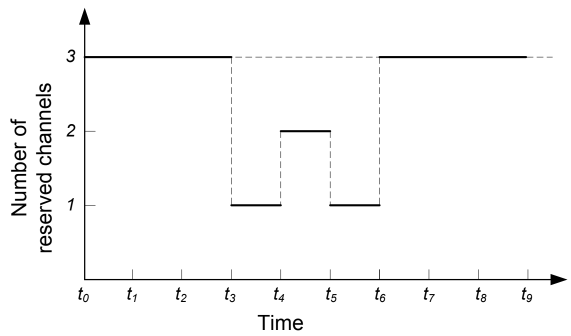
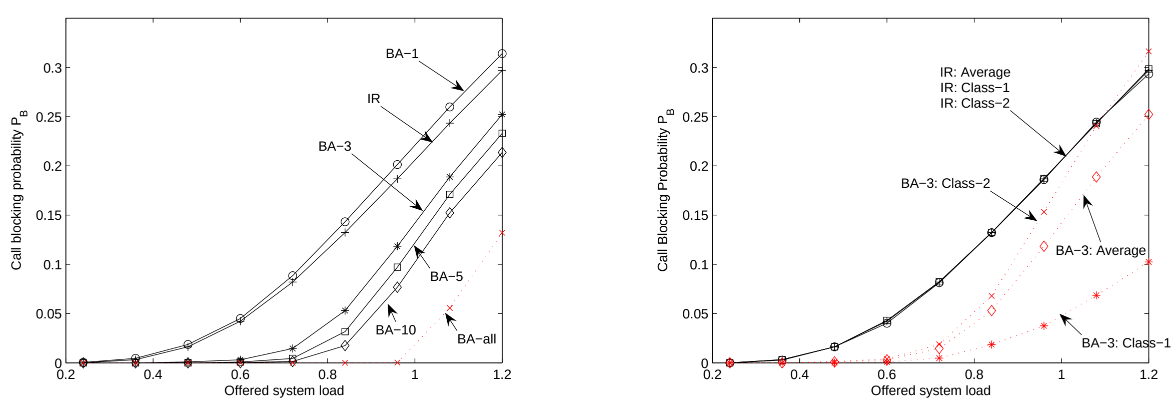


Fig. 7: Gaps in the advance-reservation horizon when call-initiation time is unrestricted.

offered system load range is maintained (24% and 120%). We do not observe any significant differences related to the plots shown in Fig. 6. The relative difference between the various BA schemes and the IR scheme stays the same.

2) Multiple call classes: Assuming two call classes, we set the call holding times, h_1 and h_2, to 100 and 300 timeslots, respectively, and set the probabilities that an incoming call belongs to class-1 and class-2, i.e., r_1 and r_2, to 0.3 and 0.7, respectively. Therefore the average call holding time is $h_1 r_1 + h_2 r_2 = 240$ timeslots. We change p from 0.01 to 0.05 to simulate the system under the same offered load range (24% to 120%) and plot the call blocking probability against offered system load in Fig. 8a.



(a) Average call blocking probability

(b) Class based call blocking probability of IR and BA−3 calls

Fig. 8: Performance of different bandwidth sharing mechanisms with two classes of calls. $m = 10$, $l = 2$, $h_1 = 100$, $h_2 = 300$, $r_1 = 0.3$, $r_2 = 0.7$, and $K = 2000$.

Unlike the results shown in Fig. 6 for the `BA-n` schemes (dashed lines), here the `BA-n` mechanisms outperform the immediate-request mechanism even when the system load is very high. Also, the performance gains of the book-ahead mechanisms are more significant than in the single class cases. The reason for this is an interesting play of effects between the two call classes (see Fig. 8b). Longer-duration calls will be blocked at a higher rate because of the reason provided in the previous subsection with regards to the assumed distribution for call-initiation times. As more longer-duration calls are blocked, shorter-duration calls enjoy a greater probability of fitting into gaps left in the reservation timeline. Thus, they enjoy a lower call blocking probability. The aggregate call blocking probability is much lower than with the IR scheme. The same phenomenon is observed even if we exchange the relative call-arrival rate factors to 70% for the 100-timeslot calls and 30% for the 300-timeslot calls.

D. *Restricted call-initiation times*

We present numerical results obtained from the simulation model when call-initiation time options in the `BA-n` schemes are restricted to fall on timeslot boundaries separated by the minimum call holding time. In these simulations, we assume only one call class. The probability that a call chooses specific restricted timeslot boundaries is still assumed to be uniformly distributed.

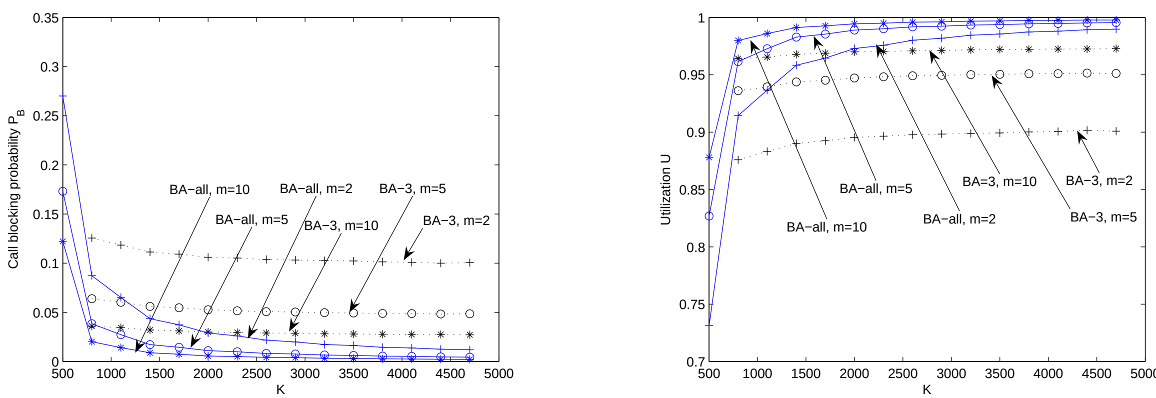
1) Improved performance of `BA-n` schemes: The solid lines in Fig. 6 shows the performance of different BA schemes under the restricted call-initiation time policy for a single class of calls with a holding time of 300 timeslots. The results for the unrestricted call-initiation time policy (from Section V-C) are also plotted in the same figure for comparison.

As seen in Fig. 6, the performance of the restricted `BA-n` mechanisms is much better than the IR mechanism. Even the `BA-1` scheme achieves a 28% reduction in call blocking probability relative to the IR scheme when the offered load is 120%. Using even just 3 options, the BA scheme performs almost as well as the `BA-all` scheme. As expected, there is no difference between `BA-all` with and without the restrictions.

Comparing the solid and dotted utilization curves in Fig. 6, we see that while the `BA-n` schemes achieve a maximum utilization of 86% when call-initiation times are unrestricted, 100% utilization is achievable in the restricted case. For example, with the `BA-3` scheme, a call blocking probability of 8% is achievable when operating the link at a 98.7% utilization with restricted call-initiation time options, while without restriction, the `BA-3` scheme only achieved a maximum of 84.8% utilization with a 28.8% call blocking probability. It is important to achieve good results with the `BA-n` schemes, rather than with just `BA-all`, because we expect scientists to want the flexibility of selecting specific timeslots in which to reserve network bandwidth for their experiments, data transfers, or remote collaborations.

2) Impact of call holding time: Assuming a single call class, we obtain numerical results for three values of the call holding time, 100, 300, and 500 timeslots. We set the call arrival probability correspondingly so that the results can be compared for the same offered system load range. We repeat the simulations for three different values of m, 2, 5, and 10. K is fixed at a factor of 15 times the call holding time, i.e., $K = 15H$. The simulation results show that the call blocking probability is independent of the call holding time for a fixed value of the offered load. However it does depend on m, whose effect we present in the next subsection.

E. Dependence of K on m and H



(a) Call blocking probability

(b) Utilization

Fig. 9: Performance of `BA-3` and `BA-all` under different values of the advance-reservation horizon,K, and number of channels, m. $l = 1$, $h_1 = 200$, and load$= 100\%$.

As stated in Section V-A, the important input parameters are m, H and K. In this subsection, we present the impact of m and H on the selection of the advance-reservation horizon, K. The latter is a parameter that needs to be explicitly selected based on m, H, and the type of BA scheme used. It is obvious that the longer the advance-reservation horizon, the better the system performance. However, the longer the advance-reservation horizon, the greater the storage and computation needs of the scheduler. Given we do not have a closed form solution for the analytical model to derive an optimum value of K, we simulate the system to gain some understanding on acceptable values for K.

Fig. 9 shows simulation results for two book-ahead mechanisms, `BA-3` and `BA-all`, for three values of m, 2, 5, and 10. Given the results of the previous subsection, which showed independence of system performance on call holding time, we simply fix the holding time to 200 timeslots. The call arrival probability p is chosen for different m so that the offered system load is fixed at 100% for all points shown in the plots. Given that the call-initiation time options are restricted, for the `BA-3` scheme, K should be at least 4 times the call holding time H to allow the user to specify three call-initiation time options. Therefore the plots for the `BA-3` schemes start at $K = 800$.

Fig. 9 shows that the call blocking probability of the `BA-all` scheme approaches 0 as the advance-reservation horizon K increases. However, the performance of `BA-3` is limited (the call blocking probability does not approach 0) even if K goes to infinity. The reason is that the limitation of 3 options overrides the performance gain caused by large K.

Fig. 9 also shows the performance improvement is small after K reaches a certain value. For example, when $m = 10$, the threshold is around 1500 for `BA-3` and 2500 for BA-all, respectively. The threshold is smaller with `BA-3` because, given the user is limited to specifying only 3 options, an increase in K does not yield further improvement. The implication of these results is that the reservation period does not need to be very long in order to reap the benefits of the book-ahead mechanism.

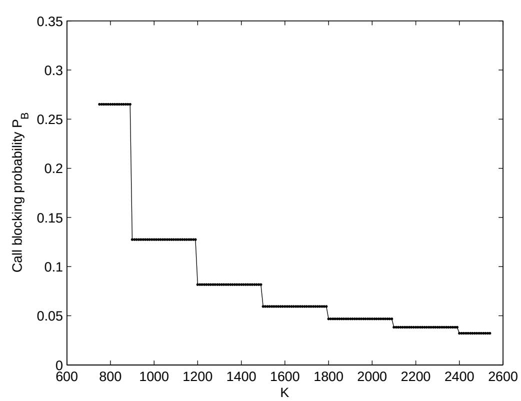


Fig. 10: Minimum K/H values: m=2, H=300, offered load=100%, `BA-all`.

TABLE II: Dependence of K/H on m: offered load=100%, `BA-all`.

Call blocking probability	2%	5%	10%
$m = 2$	14	6	4
$m = 5$	5	4	3
$m = 10$	4	3	2

We observe from Fig. 10 that call blocking probability depends on the ratio K/H rather than on K itself. This result is explained by the fact that call-initiation time options are restricted to time boundaries separated by H timeslots. Fig. 9 shows that even for a small m ($m = 2$), a K/H factor of 6 is sufficient to operate the system at a call blocking probability of 5%.

From Section V-D.2 we know that call blocking probability is dependent on m and offered load, but not independently on H. Combining this observation with that noted in Fig. 10, which shows dependence on K/H rather than K, we list K/H values for different values of m corresponding to 3 values of call blocking probability for the `BA-all` scheme in Table II. It shows that to achieve a 2% call blocking probability, if $m = 2$, we need to select a much longer advance-reservation horizon K than if $m = 10$.

VI. Conclusions

We presented a novel discrete-time Markov chain model of book-ahead bandwidth-sharing mechanisms in optical circuit-/VC-switched networks. We used this analytical model and a simulation model to understand the benefits of book-ahead (BA) bandwidth-sharing when compared to the immediate-request (IR) call-blocking mode of bandwidth-sharing in circuit-switched networks. In optical testbeds being created to support scientific research applications, the per-circuit bandwidth is high relative to link capacity. This makes m small, i.e., on the order of 1 to 10. In this range, the Erlang-B formula shows that the IR bandwidth-sharing mode cannot achieve both low call blocking probability and high link utilization simultaneously. With BA schemes, we showed that a significant improvement in call blocking probability is possible even at a high utilization.

We studied two different BA schemes, `BA-all` and `BA-n`. In `BA-all`, the caller accepts any set of available timeslots, while in `BA-n`, the caller specifies n call-initiation time options and is admitted only if a channel is available starting from one of these n options for a contiguous set of timeslots equal in length to the requested holding time. The `BA-all` mechanism achieves 95% utilization with a call-blocking probability of only 1%, while in the IR mode, call blocking probability is 23% even when utilization is only 80%. In the `BA-n` schemes, restricting the call-initiation time options to fall on timeslot boundaries separated by the minimum call holding time, yields better results. Allowing users to request any call-initiation time, without regard to call holding times, results in increased blocking. With multiple call classes, this need for restrictions on call-initiation times is less important; nevertheless call classes with longer holding times suffer higher call blocking probabilities than in the IR scheme if no such restrictions are imposed.

In a single call class system with restricted call-initiation time options, we determined that the length of the advance-reservation horizon, K, required increases linearly with the holding time H. The ratio K/H is primarily dependent on m. For example, if $m = 2$, to achieve a 2% call blocking probability, the advance-reservation horizon needs to be a factor of 14 times the call holding time, while this factor drops to 4 when $m = 10$. Thus the extra data storage and processing required to accept and maintain advance reservations is not significant. A final note is that our simulation results also show that the call blocking probability is independent of the call holding time for a given offered load.

Acknowledgment

This work was carried out under the sponsorship of NSF ITR-0312376, NSF ANI-0335190, NSF ANI-0087487, and DOE DE-FG02-04ER25640 grants.

References

[1] CANARIE's CA*net 4. [Online]. Available: http://www.canarie.ca/canet4/index.html

[2] OMNInet. [Online]. Available: http://www.icair.org/omninet/

[3] SURFnet. [Online]. Available: http://www.surfnet.nl/info/en/home.jsp

[4] UKLight. [Online]. Available: http://www.uklight.ac.uk/

[5] ESnet. [Online]. Available: http://www.es.net/

[6] TeraPaths. [Online]. Available: http://www.atlasgrid.bnl.gov/terapaths/

[7] Dynamic resource allocation via gmpls optical networks (DRAGON). [Online]. Available: http://dragon.east.isi.edu/

[8] N. S. V. Rao, W. R. Wing, S. M. Carter, and Q. Wu, "Ultrascience net: Network testbed for large-scale science applications," *IEEE Commun. Mag.*, vol. 43, no. 11, pp. 12–17, Nov. 2005.

[9] CHEETAH. [Online]. Available: http://cheetah.cs.virginia.edu

[10] E. Mannie, "Generalized multi-Protocol label switching (GMPLS) architecture," RFC 3945, Oct. 2004.

[11] L. Berger, "Generalized multi-protocol label switching (GMPLS) signaling - resource reservation protocol-traffic engineering (RSVP-TE) extensions," RFC 3473, Jan. 2003.

[12] D. Katz, K. Kompella, and D. Yeung, "Traffic engineering (TE) extensions to OSPF version 2," RFC 3630, Sept. 2003.

[13] M. Veeraraghavan, X. Fang, and X. Zheng, "On the suitability of applications for gmpls networks," in *Submitted to Globecom 2006*.

[14] X. Zheng, M. Veeraraghavan, N. S. V. Rao, Q. Wu, and M. Zhu, "CHEETAH: Circuit-switched high-speed end-to-end transport architecture testbed," *IEEE Commun. Mag.*, vol. 43, no. 8, pp. 11–17, Aug. 2005.

[15] M. Schwartz, *Telecommunication networks: protocols, modeling and analysis*. Boston, MA: Addison-Wesley, 1986.

[16] L. C. Wolf and R. Steinmetz, "Concepts for resource reservation in advance," *Multimedia Tools Appl.*, vol. 4, no. 3, pp. 255–278, 1997.

[17] D. Ferrari, A. Gupta, and G. Ventre, "Distributed advance reservation of real-time connections," in *Proc. of the Fifth International Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, NH, Apr. 1995.

[18] A. Naik, H. J. Siegel, and E. K. P. Chong, "Dynamic resource allocation for classes of prioritized session and data requests in preemptive heterogenous networks," in *Proc. of the 2001 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, June 2001.

[19] L. Yuan, C. Tham, and A. L. Ananda, "A probing approach for effective distributed resource reservation," in *Second International Workshop on Quality of Service in Multiservice IP Networks*, Milano, Italy, Feb. 2003.

[20] A. G. Greenberg, R. Srikant, and W. Whitt, "Resource sharing for book-ahead and instantaneous-request calls," *IEEE/ACM Trans. Netw.*, vol. 7, no. 1, pp. 10–22, 1999.

[21] E. G. Coffman, P. Jelenkovic, and B. Poonen, "Reservation probabilities," *Advances in Performance Analysis*, vol. 2, no. 2, pp. 129–158, 1998.

[22] J. T. Virtamo, "A model of reservation systems," *IEEE Transactions on Communications*, vol. 40, pp. 109–118, 1992.

[23] V. Paxson and S. Floyd, "Wide-area traffic: the failure of poisson modeling," in *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, New York, NY, USA, 1994, pp. 257–268.

[24] K. S. Trivedi, *Probability and statistics with reliability, queuing, and computer science applications*. Hoboken, NJ: Wiley, 2001.

[25] J. Heidemann, K. Mills, and S. Kumar, "Expanding confidence in network simulation," *IEEE Network Magazine*, vol. 15, no. 5, pp. 58–63, Sept./Oct. 2001.

[26] R. Yates and D. Goodman, *Probability and Stochastic Processes*. Hoboken, NJ: Wiley, 2004.