RESOURCE SHARING FOR BOOK-AHEAD AND INSTANTANEOUS-REQUEST CALLS[1]

Albert Greenberg[2] — AT&T Labs

R. Srikant[3] — University of Illinois

Ward Whitt[4] — AT&T Labs

February 21, 1997

Revision: September 4, 1998

[1]An abbreviated version of this paper was presented at the 15th International Teletraffic Congress in Washington, D.C., June 1997.

[2]Room A161, AT&T Labs, 180 Park Avenue, Florham Park, NJ 07932-0971; email: albert@research.att.com

[2]Coordinated Science Laboratory and the Department of General Engineering, University of Illinois, 1308 W. Main Street, Urbana, IL 61801; email: rsrikant@uiuc.edu

[3]Room A117, AT&T Labs, 180 Park Avenue, Florham Park, NJ 07932-0971; email: wow@research.att.com

Abstract

In order to provide adequate quality of service to large-bandwidth calls, such as video conference calls, service providers of integrated-services networks may want to allow some customers to book ahead their calls, i.e., make advanced reservations. We propose a scheme for sharing resources among book-ahead calls (that announce their call holding times as well as their call initiation times upon arrival) and non-book-ahead calls (that do not announce their holding times). It is possible to share resources without allowing any calls in progress to be interrupted, but in order to achieve a more efficient use of resources, we think that it may be desirable to occasionally allow a call in progress to be interrupted. (In practice, it may be possible to substitute service degradation, such as bit dropping or coarser encoding of video, for interruption.) Thus, we propose an admission control algorithm in which a call is admitted if an approximate interrupt probability (computed in real time) is below a threshold. Simulation experiments show that the proposed admission control algorithm can be better (i.e., yield higher total utilization or higher revenue) than alternative schemes that do not allow interruption, such as a strict partitioning of resources.

Key Words: Integrated-Services Networks, Book-Ahead Calls, Advance Reservation, Video Teleconferencing, Link Partitioning, Quality of Service, Loss Networks.

1. Introduction

In integrated-services networks, it is difficult to provide adequate quality of service to large bandwidth calls, such as video conference calls, without adversely affecting the network utilization. One way to alleviate this problem is to allow some customers of the network to book their calls ahead of the actual call initiation time, much like calling ahead to make a reservation at a restaurant. We refer to such calls as book-ahead (BA) calls.

It seems reasonable to require BA calls to announce their intended holding times as well as their call initiation times, and that is what we require. However, there may also be calls that do not book ahead, referred to as instantaneous-request (IR) calls, which do not announce their holding times. A natural way to allow for both BA calls (which announce their holding times) and IR calls (which do not) is to partition the resources into two disjoint subsets dedicated to each class. Without strict partitioning, the resource could also be shared without allowing any calls in progress to be interrupted, e.g., by having a moving boundary between the classes. However, we contend that it is also desirable to consider resource sharing in which some calls in progress can be interrupted. In many applications it will not actually be necessary to interrupt calls. Instead, the bandwidth or the quality of service will be reduced, e.g., by bit dropping or coarser encoding in video. While we only speak of interruptions, our admission control algorithm can be used with other forms of service degradation.

The resource sharing with the possibility of interruption that we consider is similar to admission control algorithms in wireless networks, in which small handoff dropping rates are allowed to increase the overall network utilization [14]. The possible advantage of allowing occasional call interruption or service degradation with BA calls is also similar in spirit to proposed call admission algorithms in ATM networks; instead of reserving resources to accommodate traffic at its peak rate, small cell loss probabilities are allowed in order to increase the number of sources that can be admitted [11]. It should be noted though that call interruption is usually more serious than cell dropping. However, as noted above, in practice it may be possible to substitute temporary service degradation for call interruption.

In our discussion, we act as if each call requires capacity from a single resource (link), but our admission control algorithm applies directly to networks. With a specified routing rule, such as shortest-path routing, we admit each call if the criteria are met at each required resource (e.g., link). There is also the possibility of exploiting the book-ahead feature in order to improve the routing decisions, but we do not discuss the possible interplay between routing and book-ahead here.

In our discussion, we also act as if each call requires a fixed amount of bandwidth throughout the duration of the call. (However, the fixed bandwidth requirements of BA calls can differ from the fixed bandwidth requirements of IR calls and other BA calls.) Our algorithm may also apply to calls that require variable bandwidth if we can act as if they require a fixed bandwidth, by the use of effective bandwidths; e.g., see Kelly [13] and references therein.

A major motivation for us is the existence of a commercial book-ahead service, AT&T's ACCUNET Bandwidth Manager [1]. However, it currently requires *all* calls to specify their holding times in advance. Recently, others have begun studying book-ahead service, under the name advance reservation [7, 10, 23]. (We prefer the phrase "book-ahead" to "advance reservation" in order to avoid possible confusion with trunk reservation [19].) Ferrari, Gupta and Ventre [10] consider the implementation issues in providing a BA service where the BA calls request multiparty connections. They present a way to implement a BA service using existing protocols, primarily in the framework of Tenet protocols. They do allow limited resource sharing sharing between BA and IR calls, but they do not use interrupt probabilities. (Their scheme evidently is the special case of our algorithm CHTA in Section 5.4 with $H = \infty$.) Wolf, Delgrossi, Steinmetz, Schaller and Wittig [23] discuss issues associated with providing a BA service. While both [10] and [23] allow resource sharing, they do not indicate how to treat IR calls that do not announce their holding times. Degermark, Kohler, Pink and Schelen [7] do not allow resource sharing, but they consider how to predict the amount of resources needed using past measurements.

Here is a quick summary of our proposed admission control policy: We admit an IR call if the probability of it being interrupted is below a specified threshold. (If a call must be interrupted, we assume that the most recent IR arrival is interrupted, but other interrupt policies could be used.) We assume that BA calls book far ahead relative to IR holding times, and enforce that assumption if necessary by having a minimum book-ahead time. When deciding whether or not to admit a BA call, we ignore IR calls in progress. To give IR calls protection we impose an upper limit on the number of BA calls in the system.

Here is how our paper is organized: We begin in Section 2 by discussing the formulation of the admission control problem when there are both BA and IR calls. In Section 3 we specify the traffic model and service objectives. In Section 4 we develop an efficient algorithm to describe the performance in the important special case of widely separated time scales, in which IR calls arrive and depart much more quickly than BA calls. We use that algorithm to show, analytically, that resource sharing can significantly outperform strict link partitioning.

In Section 5 we develop our admission control algorithm based on approximate interrupt probability calculations, under the assumption of an exponential IR holding-time distribution. In Section 6 we investigate the performance of this algorithm with alternative interrupt probability approximations using simulation experiments. In Section 7 we extend the algorithm to cover the case of non-exponential holding-time distributions. We primarily consider the case in which BA calls book far ahead compared to IR holding times. However, in Section 8 we consider the case in which BA calls do not book far ahead. Finally, in Section 9 we state our conclusions.

2. The Admission Control Problem

In this section we carefully formulate the admission control problem when there are both BA and IR calls. We assume that IR calls enter service immediately upon arrival if they are admitted, without announcing their holding times. In contrast, BA calls announce a proposed book-ahead time and a proposed call holding time. If a BA call is admitted, then it enters service at the original call arrival time plus the book-ahead time, it spends the call holding time in service, and then it departs. (If t_1, t_2 and t_3 are the request arrival time, book-ahead time and holding time for a BA call, then it would be in service in the interval $[t_1 + t_2, t_1 + t_2 + t_3]$ if it is admitted.)

The announced BA holding time may of course be an estimate or a safe upper bound. The capacity used by this BA call will be made available to other calls when the BA customer departs or the holding time expires, whichever happens first. A BA call might also be allowed to extend its holding time. A simple way to do this is to treat such a request as a new BA call. For this new request, the book-ahead time would be the interval between the request epoch and the epoch the BA call was previously scheduled to depart. The holding time of the new request would be the incremental holding time. (In the setting above with times t_1, t_2 and t_3, if the BA call made a request at time t_4 to depart at time $t_1 + t_2 + t_3 + t_5$, where $t_1 < t_4 < t_1 + t_2 + t_3$, then the second request would have book-ahead time $t_1 + t_2 + t_3 - t_4$ and holding time t_5.)

We are primarily interested in the case in which BA calls book far ahead compared to IR holding times. For example, in a standard telecommunications network, ordinary IR voice calls have a mean holding time of a few minutes, while teleconference calls may be booked ahead hours or even days in advance. Hence, here we assume that BA calls do book far ahead, except in Section 8. If necessary, this condition can be enforced by having a minimum book-ahead time. Under this condition, when considering whether or not to admit a BA call, the IR calls in progress need not be considered. A BA call is admitted (scheduled in the future) if there is room for it considering only previously booked

BA calls.

In this setting, the main problem is to determine an admission control policy for IR calls. We propose an admission control policy for IR calls based on an interrupt probability computation for each arriving call. Our policy lets an IR call be admitted if a computed interrupt probability is less than a certain threshold; otherwise the call is blocked.

It remains to specify which call will be interrupted when there is contention. We are thinking of interruptions as rare events, so that the specific choice of which call to interrupt should not effect the performance of the algorithm much. Hence it is natural to interrupt the least valuable call, whatever that happens to be. If the value of a completed call increases with its duration, then it is better to interrupt a call that arrived more recently. With that case in mind, we assume that the most recently admitted IR call is interrupted if an interruption is necessary.

We should emphasize that the long-run proportion of calls that are actually interrupted will usually be substantially lower than the interrupt probability threshold, because the threshold is only an upper bound on the calculated interrupt probability for each call. Since BA calls book relatively far ahead and the policy is to interrupt the most recent IR arrival, future events will not alter the interrupt probabilities computed upon IR call arrival. Hence, the threshold is an upper bound. We use simulation to determine the long-run proportion of calls that are actually interrupted with a given threshold.

Since BA calls book relatively far ahead, the book-ahead feature gives BA calls priority over IR calls. Thus it may be desirable to also provide some service guarantees to IR calls. For this purpose, we propose using an *upper limit* on BA calls to control the admission of BA calls. In this context, the upper limit is equivalent to the popular *trunk reservation* control; i.e., a BA call will not be admitted if the spare capacity (considering only BA calls) after admitting the call is less than some trunk reservation parameter r at all times in the future. Since IR calls are not considered when applying trunk reservation against BA calls, the control is equivalent to an upper limit on the number of BA calls in the system.

In our simulations we assume that arriving BA calls whose initial requests cannot be met are blocked and lost. However, in reality, the BA calls could modify their requests, i.e., accept an alternative available time slot. It is significant that our admission control policy for IR calls applies equally well with such modifications. In our simulations we assume that, upon the arrival of each call, the service provider knows the number of IR calls in progress and the number of previously admitted BA calls (in progress or scheduled) that will be present at all times in the future, which we refer to as the *BA call profile*. What is computed (approximately) is the probability of an interruption at any time in the future, given the calls in service, the previously scheduled BA calls and the new arrival, but ignoring all future arrivals.

To implement such an admission control policy, it is important to ensure that the interrupt probability computation is fast, so that the admission control decision can be made in *real time*. Since the exact computation of the interrupt probability can be computationally prohibitive, we propose several approximate schemes for this computation. Through simulation, we show that these approximation schemes are effective from the perspective of both real-time computation and expected revenue. In particular, we show that the proposed admission control algorithm can yield more revenue than alternative schemes that do not allow interruptions.

Since some IR customers may object strongly to interruptions, it may be desirable to have multiple classes of IR calls, only some of which can experience interruptions. A scheme for providing multiple grades of service for multiple customer classes is described in Choudhary, Leung and Whitt [3]. That scheme provides resource sharing with protection against overloads through the use of guaranteed minimum and upper limit bounds. With multiple classes, it might be decided to interrupt the most recent arrival from the lowest ranked class present. Instead, a more complicated algorithm might be considered, in which both class type and arrival time play a role. In this paper we only discuss a single

IR customer class, but our approach extends to multiple IR classes, provided that the IR classes that may experience interruption have a common holding time distribution.

In this paper the BA calls can have very general (constant) bandwidth requirements, book-ahead times (time until starting service) and holding times (service durations); there can even be multiple BA classes. However, throughout this paper we assume that there is a *single class* of IR calls with unit bandwidth requirement, common holding-time distribution and common performance requirements. Since the present paper was completed, other admission control algorithms have been proposed by Wischik and Greenberg [22] and Srikant and Whitt [21] that allow multiple classes of IR calls. The algorithm in [22] is based on effective bandwidths, exploiting large deviations analysis, while the algorithm in [21] exploits the central limit theorem. In addition to allowing multiple IR classes, the scheme in [22] does not require that BA calls specify their holding times. Preliminary investigations indicate that the admission control algorithms here and in [21] are more effective for the narrower class of problems to which they apply.

3. Traffic Model and Service Objectives

In this section we present our traffic model and service objectives, which we use to evaluate the performance of our proposed admission control algorithm. We assume that BA and IR calls (service requests) arrive according to independent stationary stochastic point processes with rates λ_B and λ_I. We assume that the BA (IR) call holding times have a common distribution with mean μ_B^{-1} (μ_I^{-1}). We assume that the successive BA book-ahead times are i.i.d. random variables with mean t_b. We assume that the arrival processes, holding times and book-ahead times are all mutually independent. We also assume that IR calls request 1 unit of bandwidth, BA calls request b unit of bandwidth and the total available bandwidth on the link is s. (Our approach extends to heterogeneous BA calls with different bandwidth requirements, but it exploits the common bandwidth requirement for IR calls, which need not be 1 unit.) We assume that the most recent IR arrival is interrupted when an interruption is necessary.

The performance of the proposed admission control policy primarily depends only on the assumptions about the IR holding times. In Sections 5 and 6, we assume that the IR holding times are exponentially distributed. The admission control algorithm based on exponential holding times with mean μ_I^{-1} also performs reasonably well if the mean is not μ_I^{-1} (but is not drastically different) or if the distribution is not exponential, but performance can be improved by taking into account the true distribution. For this purpose, we also develop an algorithm for computing interrupt probabilities with a general IR holding-time distribution in Section 7. In this case, we make a further assumption that the IR calls arrive according to a Poisson process. This non-exponential-distribution algorithm has the same computational complexity as in the exponential case. It would be natural to also use this algorithm as an approximation for other, non-Poisson, arrival processes. With or without a Poisson arrival process, the elapsed holding times (ages) of the IR calls in progress have an impact on the residual holding-time distribution when the underlying holding-time distribution is not exponential, but our algorithm in Section 7 does not use the ages. (See [21] and [22] for alternatives that do.)

To characterize the performance of the admission control, we focus on the following:

- P_I: Blocking probability for IR calls, i.e., the long-run fraction of IR calls that are rejected either by the admission control algorithm or due to the link being full.
- P_B: Blocking probability of BA calls, i.e., the long-run fraction of BA calls that are rejected, either due to insufficient capacity for the entire duration of the pre-announced holding time or due to some admission control such as an upper limit.

- p_I: The interrupt probability for IR calls, i.e., the long-run fraction of admitted IR calls that are interrupted while they are in progress due to the link being full.

As indicated above, we assume that BA calls are not interrupted. When there is contention among calls in progress, we assume that IR calls are interrupted, with the most recent IR arrival being interrupted first.

In this context, performance as described by the three characteristics P_I, P_B and p_I is determined by three controllable factors: the overall capacity, the upper limit on BA calls and the interrupt-probability threshold. Since the BA calls book relatively far ahead, the BA blocking probability P_B depends only upon the capacity and the BA upper limit, not upon the IR interrupt-probability threshold, denoted by P_T. Indeed, we can determine P_B by separately considering only the BA calls.

For given total capacity, BA upper limit and traffic characteristics, the IR-call performance characteristics P_I and p_I depend on the IR interrupt-probability threshold. As this threshold decreases, we will tend to take less of a chance on admitting IR calls; i.e., p_I will go down, while P_I will go up. For the IR calls, there is an important tradeoff between p_I and P_I.

It is possible to formulate the admission control problem as an optimization problem, so that the goal becomes maximizing net revenue. Once costs and benefits are specified, we can use simulation to help determine the three controls (total capacity, upper limit for BA calls and IR interrupt-probability threshold) that yield maximum net revenue. More generally, once the criterion has been specified, we could attempt to do even better by considering other kinds of admission control policies, but we do not.

To illustrate the optimization approach, suppose that there are *per-call* revenues of R_I and R_B and *per-time* revenue rates of r_I and r_B for IR and BA calls that complete service, and a *per-call* cost of C_I for interrupting an IR call, Then the admission control scheme for admitting IR and BA calls can be chosen to maximize the rate of revenue

$$\mathcal{R} \equiv (1-P_I)(1-p_I)(R_I + \frac{r_I}{\mu'_I})\lambda_I + (1-P_B)(R_B + \frac{r_B}{\mu_B})\lambda_B - p_I(1-P_I)C_I\lambda_I, \tag{3.1}$$

where $1/\mu'_I$ is the average holding time for IR calls that are admitted and *not* interrupted. It is important to note that μ'_I is not μ_I; conditioning on not being interrupted affects the holding-time distribution. Indeed, experience shows that the average completed portion of interrupted calls tend to be greater than $1/\mu_I$, whereas the average length of uninterrupted calls tends to be less than $1/\mu_I$. However, when the interrupt probability is very small, as is usually desired, then μ'_I tends to be nearly the same as μ_I. Hence, one might substitute μ_I for μ'_I in (3.1).

Alternatively, the admission control scheme for admitting IR calls could be chosen to maximize the rate of revenue

$$\mathcal{R} \equiv (1-P_I)(1-p_I)(R_I + \frac{r_I}{\mu'_I})\lambda_I + (1-P_B)(R_B + \frac{r_B}{\mu_B})\lambda_B \ , \tag{3.2}$$

subject to the constraint $p_I \le P$ where P is an upper bound on the long-run interrupt probability of IR calls. Alternatively, the constraint could be expressed by the interrupt probability threshold for individual calls. In this paper, we use the formulation (3.2), but our framework also allows the use of (3.1). Indeed, our framework allows for many alternative revenue functions. For any one, we can apply simulation to determine desirable settings for the three controls.

In our examples, we make the parameter choice $R_I = R_B = 0$ and $r_I = r_B = 1$ which makes the revenue correspond simply to utilization. While it is natural to focus on utilization, it is also of interest to consider alternative pricing schemes. Due to the impact of larger-bandwidth BA calls on IR calls, it might be thought that we should have $r_B > r_I$. On the other hand, volume discounts might dictate $r_B < r_I$. We do not examine such alternatives here, but we provide a basis for studying them.

4. An Algorithm for Separated Time Scales

In this section we show through a relatively simple example that resource sharing can be superior to strict link partitioning. The advantage of sharing is well known for loss models without booking ahead. Indeed, link partitioning tends to be effective only in special traffic regimes, such as heavy traffic [19, Chapter 4.2], [3].

In order to obtain an analytically tractable regime, we consider the situation of widely separated time scales, in which $\lambda_I >> \lambda_B$ and $\mu_I >> \mu_B$. This is a regime commonly occurring in multimedia networks, in which voice calls arrive and depart more frequently than large bandwidth calls such as video, e.g., see [8]. Let BA calls book far ahead and give them priority over IR calls when there is resource contention.

The separation of time scales between the two classes allows us to develop an efficient numerical algorithm to describe the performance. First, assuming that the two arrival processes are Poisson processes and that the holding times come from independent sequences of independent and identically exponentially distributed random variables, the system can be described exactly by a two-dimensional Markov Chain indicating the number of BA and IR calls in the system, where neither call class books ahead but with the BA calls having preemptive priority. Second, because of the time-scale difference between the two call classes, this Markov chain is nearly decomposable [4, 15]. In particular, we can ignore the IR calls when considering the BA calls and, when we consider the IR calls, we can act as if there is a fixed amount of capacity used by BA calls. Thus, we can use the steady-state distribution for the IR calls, ignoring fluctuations of the BA calls. In other words, instead of analyzing a difficult a two-dimensional Markov chain, we can do several analyses of a simple one-dimensional Markov chain, which corresponds to the classical Erlang loss model.

In summary, we can calculate an approximation for the steady-state distribution of the two-dimensional Markov chain as follows:

- First compute the steady-state occupation probabilities of the slow-time-scale BA class using the well-known results for $M/M/\tilde{s}/\tilde{s}$ systems [2], where $\tilde{s} \equiv (s-r)/b$, and r is the trunk reservation parameter that limits the maximum number of BA calls to $(s-r)/b$. (For simplicity, assume that s and r are multiples of b.) Let $p(x_B)$ denote the steady-state probability that there are x_B BA calls in the system. The steady-state blocking probability for BA calls is $p(s/b)$.

- Next, compute the steady-state blocking probabilities for the fast-time-scale IR calls conditional on each level x_B of BA calls. In other words, compute $B_E(s-x_B, \lambda_I/\mu_I)$, where $B_E(C,\rho)$ denotes the Erlang-B formula [2] for a system with capacity C and offered traffic ρ.

- Finally, compute the steady-state blocking probability for IR calls by taking the average of the results in the second step weighted by the probabilities in the first step, i.e., by $\sum_{x_B=0}^{s/b} B_E(s - x_B, \lambda_I/\mu_I)p(x_B)$.

To consider a concrete example, let $s = 100$, $r_B = r_I = C_I = 1$, $R_B = R_I = 0$, $b = 10$, $\lambda_B/\mu_B = 2$, and $\lambda_I/\mu_I = 60$. With these costs and rewards, total revenue corresponds simply to utilization. In the blocking probability computation for nearly-decomposable systems above, the values of λ_B, μ_B, λ_I and μ_I enter only through the ratios λ_B/μ_B and λ_I/μ_I. The resulting blocking probabilities are $P_B = 0.000038$ and $P_I = 0.0283$. Since we assume that $\mu_B << \mu_I$ and $\lambda_B << \lambda_I$, we can conclude that p_I will be negligible. (Note that IR calls can only be interrupted at times when the number of BA calls in the system increases and BA arrivals are relatively infrequent in the time scale of IR calls. We would use the FTA algorithm in the next section to further reduce the IR interruptions.) Table 1 presents the the values of the revenue $\mathcal{R}$ as a function of r. Here $\mathcal{R} = 80 - 60P_I - 20P_B$.

r	P_B	P_I	R
0	.000038	.028	78.30
10	.00019	.028	78.30
20	.00086	.028	78.30
30	.0034	.027	78.28
40	.012	.026	78.21

Table 1: Revenue Under Full Sharing.

For the case of link partitioning, let us denote the capacity allocated to BA calls by s_B. The values of $\mathcal{R}$ as a function of s_B is provided in Table 2.

s_B	P_B	P_I	R
10	0.666	0.0006	66.67
20	0.400	0.0022	71.87
30	0.211	0.0237	74.34
40	0.095	0.0963	72.34

Table 2: Revenue Under Link Partitioning.

In this example, full sharing performs substantially better than link partitioning. The advantage of full sharing over link partitioning looks more dramatic if we consider lost revenue $(80 - R)$. With full sharing, the lost revenue is 1.7; with link partitioning, it is 5.6. Therefore, in general, it is worthwhile to consider admission control mechanisms other than link partitioning.

Example 4 in Section 6 shows that the nearly-decomposable Markov-chain (ND-MC) analysis here is substantiated by simulation for the specific case $\mu_I = 1$ and $\mu_B = 0.1$. That example shows that the ND-MC analysis can provide a useful approximation even when the time scales are only moderately separated.

5. Admission Control Schemes for IR Calls

As we show below, it is possible to give an expression for the interrupt probability (considering only previously accepted BA calls), but actually performing the calculation can be difficult. Thus we propose implementing our admission control strategy by calculating an *approximation* for the interrupt probability. We present several candidate approximation schemes below. The most promising one seems to be the *independent peaks approximation* (IPA), developed in Section 5.2 below, which provides high performance at manageably low computational overhead. In this section we assume that the IR holding-time distribution is exponential; in Section 7 we treat the case of a general IR holding-time distribution.

5.1. Computing the Interrupt Probability

We now indicate how to compute the interrupt probability for each arriving IR call using the number of IR calls present at the arrival instant and the future profile of BA calls. The decision depends on whether the interrupt probability is greater than or less than a certain threshold. It will be apparent from the expressions for the exact interrupt probability computation that it is difficult to implement. This is overcome by approximations presented in Section 5.2.

Let $I(t)$ denote the number of IR calls and $B(t)$ the number of BA calls in progress at time t, respectively. Suppose that a new IR call arrives at time T_0 and a decision has to be made whether or

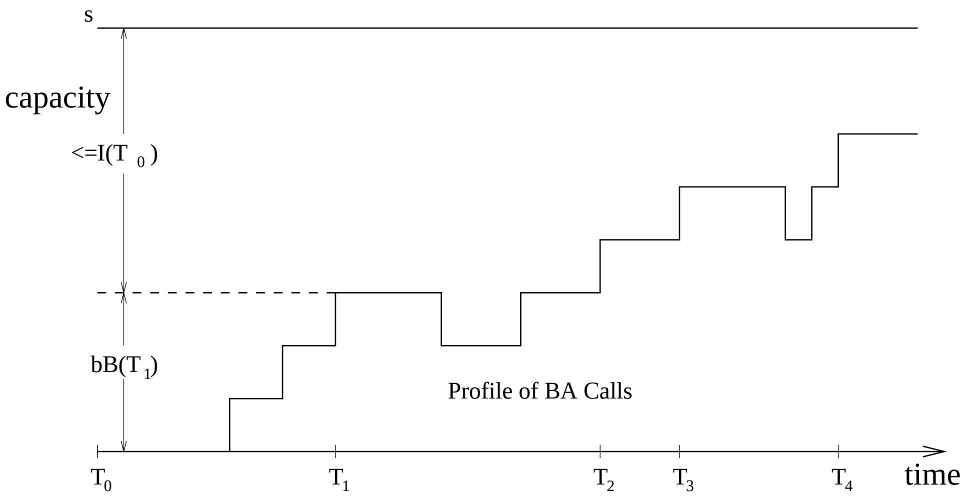

Figure 1: Interrupt times $\{T_1, T_2, T_3, T_4\}$ for an IR call arriving at T_0

not to admit this call. Then the first potential interruption time for this IR arrival at time T_0 is

$$T_1 = \min\{t \geq T_0 | I(T_0) + bB(t) \geq s\}, \tag{5.1}$$

where s is the total bandwidth (capacity). Then subsequent potential interruption times for the IR arrival at time T_0 are the times T_i, $i = 2, \ldots$, such that $T_1 < T_2 < \ldots$, and $B(T_i) = B(T_{i-1}) + 1$. An example of a BA call profile and the potential interruption times $\{T_0, T_1, \ldots\}$ is shown in Figure 1.

Let $n(T_0)$ denote the number of IR calls that have to clear down (complete their service) by T_1 so that the new call that arrived at T_0 is not interrupted. Clearly, $n(T_0) = I(T_0) - s + bB(T_1) + 1$. The number of potential interrupt times T_i is less than or equal to s/b. In practice, we can restrict attention to only those that lie within a certain interval. Suppose that we consider the first ℓ possible interrupt times, i.e., $\{T_1, T_2, \ldots, T_\ell\}$. Let $N(t)$ denote the number of IR calls (not including the new arrival at T_0) that are in the system at time T_0 and complete their service by time t for $t > T_0$. Let X denote the holding time of the new IR arrival at time T_0. Then the probability that the arriving IR call at T_0 will be interrupted at a later time, denoted by $p(T_0)$, is

$$\begin{aligned} p(T_0) &= P\Big(N(T_1) < n(T_0)\Big)\mathrm{P}\Big(X > T_1 - T_0\Big) \\ &+ P\Big(N(T_1) \geq n(T_0), N(T_2) < n(T_0) + b\Big)P\Big(X > T_2 - T_0\Big) + \ldots + \\ & P\Big(N(T_1) \geq n(T_0), \ldots, N(T_{l-1}) \geq n(T_0) + (l-2)b, N(T_l) < n(T_0) + (l-1)b\Big)P\Big(X > T_l - T_0\Big) \; . \end{aligned} \tag{5.2}$$

Assuming that IR holding times are exponentially distributed, it is possible to compute (5.2); e.g., then $P(X > t) = e^{-\mu_I t}$ and $N(t)$ has a binomial distribution with parameters $I(T_0)$ and $1 - e^{-\mu(t-T_0)}$. Even though we can give an explicit expression for (5.2), the computation is challenging.

5.2. Approximate Computations of the Interrupt Probability

We now introduce three, successively more complex, approximations for the interrupt probability in (5.2). Of course, no computation is necessary if $T_1 = \infty$.

Fixed-Time Approximation (FTA): In FTA, the interrupt probability is approximated by $p(T_0) \approx$

$e^{-\mu_I(T_1-T_0)}$; i.e., FTA ignores all interrupt times other than T_1 and does not use the information about the number of existing IR calls at time T_0.

One-Peak Approximation (OPA): The OPA uses a *lower bound* for $p(T_0)$ by using only the first term on the RHS of (5.2). In other words, we act as if there is only one interrupt time T_1. Thus, the interrupt probability $p(T_0)$ is approximated by

$$p(T_0) \approx \sum_{k=0}^{n(T_0)-1} \binom{I(T_0)}{k} \left(1 - e^{-\mu_I(T_1-T_0)}\right)^k e^{-\mu_I(T_1-T_0)(I(T_0)-k+1)} . \tag{5.3}$$

Independent-Peaks Approximation (IPA): The IPA assumes that the probability of the arriving call at T_0 being interrupted at each of the possible interrupt times $\{T_1, T_2, \ldots, T_l\}$ are independent of each other. Thus, the interrupt probability $p(T_0)$ is *upper bounded* by

$$\begin{aligned} p(T_0) &\approx \textstyle\sum_{i=1}^{\ell} \mathrm{P}\Big(N(T_i) < n(T_0) + (i-1)b\Big)\mathrm{P}\Big(X > T_i - T_0\Big) \\ &= \textstyle\sum_{i=1}^{\ell} \sum_{k=0}^{n(T_0)+(i-1)b-1} \binom{I(T_0)}{k}\Big(1 - e^{-\mu_I(T_i-T_0)}\Big)^k e^{-\mu_I(T_i-T_0)(I(T_0)-k+1)} \end{aligned} \tag{5.4}$$

We propose IPA as our preferred approximation, because we have found it to be most accurate, and it still is a very manageable computation. If we are concerned about interruptions, then IPA is conservative, because it is an upper bound. Since OPA is a lower bound, we can be sure both OPA and IPA are accurate if they are close together. A better lower bound is the maximum of the probabilities in (5.4); it is discussed in [21].

5.3. Efficiently Processing Queries to the Call Profile

In order to calculate interrupt probabilities, we need to determine the height of the BA call profile over specified intervals in the future. For moderate-sized models, this can be done in a straightforward manner, but for larger models it is desirable to have a more efficient algorithm. We outline one such algorithm now. It turns out that the queries to the call profile can be carried out in $O(\log N)$ time, where N is the number of calls in the profile. The main insight is to adapt known techniques for processing queries involving overlapping intervals, developed by computational geometers and used primarily for visibility computations [17, 16]. We intend to present the details of the query processing elsewhere. Briefly, the arrival and departure times for calls belonging to the departure time are kept in the leaves of a balanced binary search tree. Associated with each internal node of the tree is the time interval spanned by the arrivals and departures in the leaves of the subtree rooted at this node, as well as certain other information that allows us to construct the maximum height of the profile over this interval while walking the path from the root to this node. Updates to the call profile, adding newly accepted calls and dropping completed calls, trigger $O(\log N)$ time updates to this data structure.

5.4. Constant Holding Time Approximation (CHTA)

We conclude this section by introducing an admission control scheme that need not be regarded as an interrupt probability calculation. The CHTA admission control scheme for IR calls acts as if each new IR call has constant holding time H. The call is admitted if there is sufficient spare capacity in the link for H time units. Otherwise, the call is rejected. We keep track of scheduled completion times. If the call departs before H units of time, then this space is made available to any other call that requests it. Otherwise, whenever a new call arrives, we count all existing IR calls that have lasted for more than H time units as being there at this instant, but leaving a short (infinitesimal) interval later.

If we make H very large, e.g., $H = \infty$, then we essentially rule out call interruptions. IR arrivals will not be admitted if any contention is possible in the future. Similarly, new BA requests will not be admitted if IR calls in progress could then be interrupted. Evidently, Ferrari et al [10] have considered the CHTA scheme with $H = \infty$.

Note that, FTA is *not* the same as CHTA with $H = K$, because CHTA has scheduled completion times for calls in progress. We show later through simulation examples that CHTA performs poorly compared to the schemes that are based on an interrupt probability computation when the IR holding times are not in fact constant. Moreover, for finite values of H, there is "book-keeping" involved in updating the free capacity in the system.

6. Simulation Results

In this section, we present simulation results illustrating how the IR interrupt probability approximations perform. For these simulation experiments, we assume that IR and BA calls arrive according to independent Poisson processes and that all holding times are exponentially distributed. As indicated earlier, here we assume that all the BA calls book far ahead compared to IR holding times. For simplicity, we assume that all BA calls book ahead by a constant amount t_b with $t_b >> 1/\mu_I$; in particular, we let $t_b = 20$. Given that $t_b \gg 1/\mu_1$, booking ahead by a constant amount is without much loss of generality, because the BA service initiation times form a Poisson process even with random book-ahead times. (This is equivalent to the departure process in an M/GI/∞ queue being a Poisson process [9].) The constant book-ahead times ensure that the BA calls all book far ahead. This could also be achieved with a random book-ahead time that is required to exceed some minimal value.

Example 1: Let the available capacity (number of servers) be $s = 100$, the bandwidth (the number of servers) requested by each BA call be $b = 10$, the IR arrival rate be $\lambda_I = 60$, and the BA arrival rate be $\lambda_B = 2$. Let the average holding times for both call classes be 1. Recall that the bandwidth of the IR calls is assumed to be 1. Note that the offered load is $\lambda_B\mu_B^{-1}b + \lambda_I\mu_I^{-1} = 20 + 60 = 80$, so that we are in a "normal loading" regime. Without the book-ahead feature, the blocking probabilities for the BA and IR calls are 0.151 and 0.0112, respectively, as can be determined by the Kaufman [12]-Roberts [18] recursion or numerical inversion [3]. Hence, we might elect to allow booking ahead to reduce the high blocking experienced by the BA calls. When we do allow booking ahead, the BA blocking drops to the very low value 0.000038. If we allow no interruptions of IR calls, then the IR blocking probability increases to $P_I = 0.264$. (To strike a better balance, we are thus motivated to introduce the upper limit on BA calls considered in Example 3 below.)

We conducted simulations to estimate the the IR blocking and interrupt probabilities as a function of the IR interrupt probability threshold P_T. For this first example, we do not impose an upper limit on BA calls, so that $r = 0$. Given the input control P_T, we obtain estimates of P_I and p_I from each simulation run. We plot curves of P_I versus p_I based on 10 different values for P_T. For each value of P_T, the simulation run length was $100,000$ time units, after deleting 25 time units to get rid of transients. Thus, we simulate roughly 6 million IR arrivals and 2 million BA arrivals for each point in the curve. This choice of simulation run length was based on preliminary experiments, which revealed that the confidence intervals are suitably small. (The steady-state simulation run length of 100,000 was divided into 20 batches to compute confidence intervals.) The statistical accuracy is confirmed in the smoothness of the plotted curves. We also exploited our previous experience studying required simulation run lengths in loss models [20].

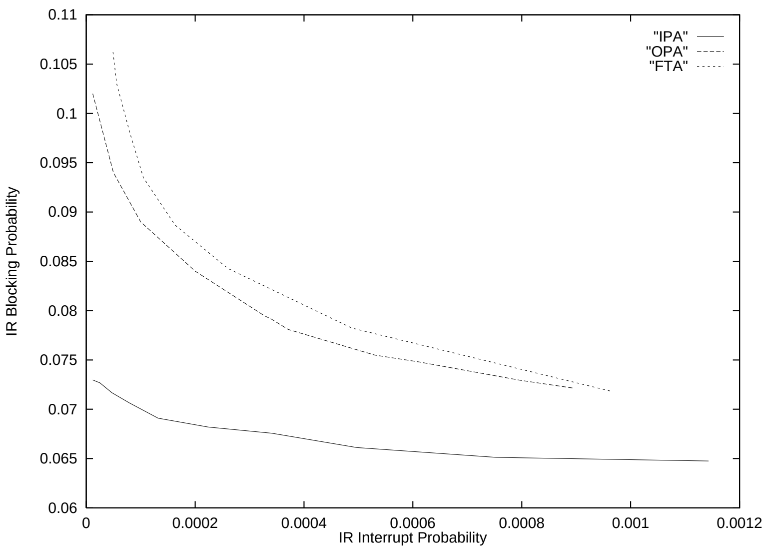

Figure 2: Plots of the interrupt probability p_I versus blocking probability P_I in Example 1. The parameters are $s = 100$, $b = 10$, $\lambda_I = 60$, $\lambda_B = 2$, $\mu_I = \mu_b = 1$.

Plots of p_I versus P_I for the three approximation procedures in Section 5.2 are shown in Figure 2. The 10 interrupt-probability threshold values were chosen between 0.01 and 0.1. Notice that P_T is very different from the realized interrupt probability p_I. The reason for this is that admitted calls may have an interrupt probability that is much smaller than P_T. Thus, on the average, the realized interrupt probability tends to be much smaller than P_T. Further, the same value of P_T yields different values for the pair (p_I, P_I) for the different admission control algorithms. For example, when P_T is 0.01, FTA results in $(0.000962, 0.071852)$, IPA results in $(0.001143, 0.064753)$, and OPA results in $(0.000893, 0.072155)$. However, for all algorithms, as P_T increases, p_I increases and P_I decreases.

The first conclusion to draw from Figure 2 is that allowing very small interrupt probabilities (e.g., of order 10^{-3}) significantly reduces the IR blocking probability (e.g., from 0.264 to under 0.07). The second conclusion is that, for each fixed p_I, IPA has significantly smaller P_I than FTA or OPA. This implies that IPA gives higher rate of revenue under both criteria (3.1) and (3.2). The algorithm CHTA in Section 5.4 performs significantly worse than the other three algorithms. Its performance is so much inferior that it is difficult to show it on the same graph with the other three algorithms without obscuring relevant details. Therefore, we present its performance in Table 3.

H	P_I	p_I
0.1	0.0194	0.03641
1.0	0.0197	0.03618
4.0	0.1135	0.00252
5.0	0.1412	0.00082
10.0	0.24	0.00005
∞	0.26	0

Table 3: Performance of CHTA for $s = 100$, $b = 10$, $\lambda_I = 60$, $\lambda_B = 2$, $\mu_I = \mu_B = 1$.

Example 2: We now consider a different scenario. We now have more peaks in the BA call profile and higher IR blocking probabilities. In particular, the parameters are $s = 40$, $\lambda_I = 24$, $\mu_I = 1$, $\lambda_B = 16$, $\mu_B = 4$, and $b = 5$. Plots of p_I versus P_I are shown in Figure 3. Each curve is based on 10 different values of P_T. Again, IPA has the best performance, but in this case the FTA curve is quite

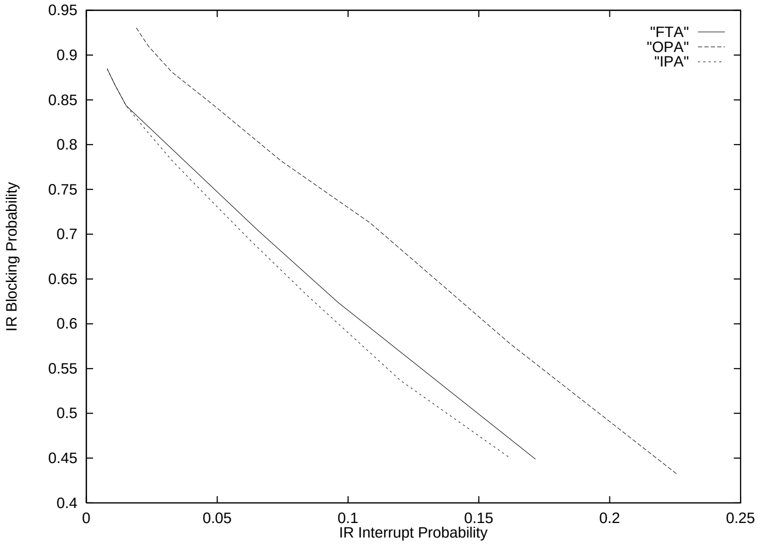

Figure 3: Plots of the interrupt probability p_I versus blocking probability P_I in Example 2. The parameters are $s = 40$, $b = 5$, $\lambda_I = 24$, $\mu_I = 1$, $\lambda_B = 16$, $\mu_B = 4$.

r	P_B
0	0.000038
10	0.000191
20	0.000859
30	0.003441
40	0.012085
50	0.036697
60	0.095238
70	0.210526

Table 4: P_B as a function of r with $s = 100$, $b = 10$, $\lambda_I = 60$, $\lambda_B = 2$, $\mu_I = \mu_B = 1$.

close to the IPA curve.

Example 3: In this example, we show how the upper limit of $s - r$ against BA calls can be effectively used to improve the rate of revenue. Consider the same set of system parameters as in Example 1, except for a new control variable r. The values of P_B as a function of r are shown in Table 4. Plots of p_I versus P_I for various values of r using the IPA policy are shown in Figure 4. As in Example 1, the curve is based on ten different values for P_T, and for each value of P_T, the simulation run length was $100,000$ after deleting 25 time units to get rid of transients.

As expected, for larger values of the reservation parameter r, both the blocking and interrupt performance of IR calls are better, at the expense of increased BA call blocking. We also plot p_I versus the average revenue $\mathcal{R}$ in Figure 5 for several values of r, assuming that we use equation (2.2) with $r_B = r_I = 1$ and $R_B = R_1 = 0$. The best results, $\mathcal{R} = 76.3$, are achieved first by $r = 50$ and then $r = 60$. In contrast, the best possible revenue with $r = 0$ is 76.0 and with link partitioning is 74.3 (with capacity 30 dedicated to BA). Since $r_B = r_I = 1$, the revenue in this example corresponds to the carried load. Since the offered load is 80, the lost revenue has been reduced from 5.7 with link partitioning to 3.7 with resource sharing using $r = 50$, a decrease of 35%. However, the effect of r alone on revenue is not great; the upper limit r is most useful for making desired tradeoffs between IR and BA performance.

We also point out that, from Figure 4, the blocking probability for IR calls is less than 0.065 when

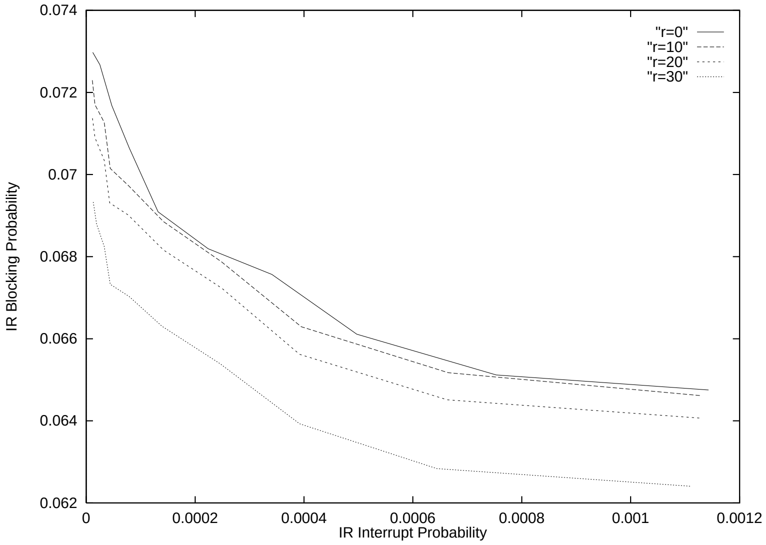

Figure 4: Plots of the interrupt probability p_I versus blocking probability P_I for various values of the upper limit parameter r in Example 3. The parameters are $s = 100$, $b = 10$, $\lambda_I = 60$, $\lambda_B = 2$ and $\mu_B = \mu_I = 1$.

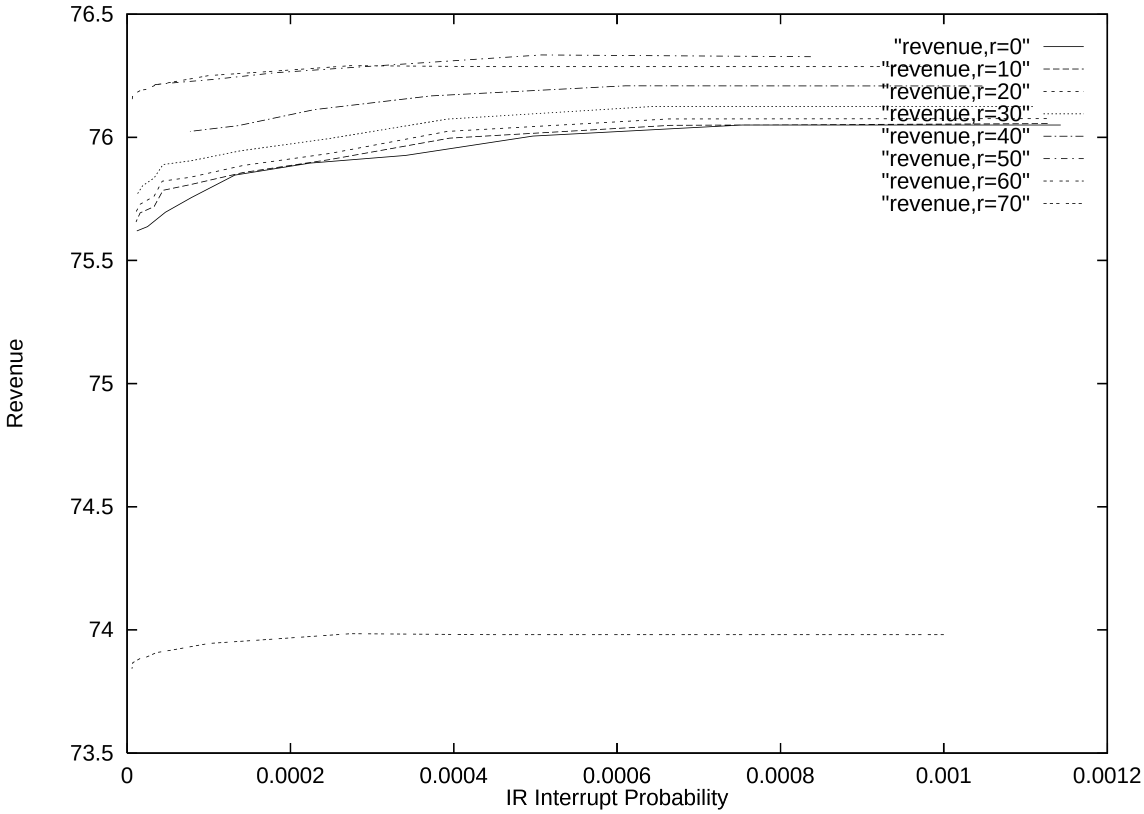

Figure 5: Plots of interrupt probability p_I versus revenue $\mathcal{R}$ in Example 3. The parameters are $s = 100$, $b = 10$, $\lambda_I = 60$, $\lambda_B = 2$, $\mu_I = \mu_B = 1$ and various values for r.

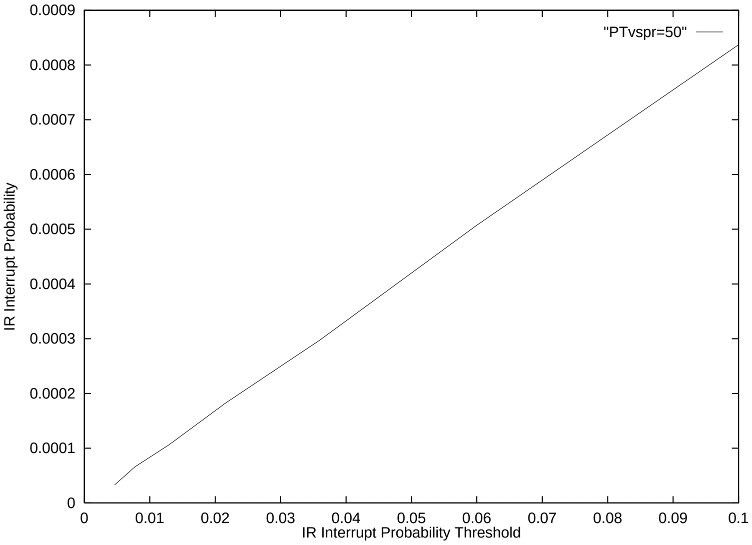

Figure 6: P_T versus p_I for Example 3 with $r = 50$

$r > 20$ and the blocking probability for BA calls is 0.037 when $r = 50$. Comparing this to Table 2, we see that our admission control algorithm can achieve high link utilization while keeping P_B and P_I reasonable whereas, under link partitioning, to achieve maximum revenue, P_B becomes 0.211 which is very high.

From a design point of view, one more relationship has to be specified to run the network at a given operating point on the p_I versus R curve. This is the relationship between interrupt probability threshold P_T and the realized interrupt probability p_I. For the optimal reservation parameter $r = 50$, this relationship is shown in Figure 6. In several examples we have found the relation between p_I and P_T to be very nearly linear. Moreover, p_I tends to be about two orders of magnitude smaller than P_T (by a factor of about 10^2).

Example 4: As mentioned in Section 1, a natural application for a book-ahead service is one where most BA calls are conference calls. Thus, it is natural to consider an example where BA calls tend to have much longer holding times than regular voice calls. Let us reconsider Example 3 with $\lambda_B = 0.2$ and $\mu_B = 0.1$. We keep the traffic intensity (λ_B/μ_B) the same but have increased the holding times of BA calls ten times. We have plotted the revenue R as function of p_I for various values of the reservation parameter r in Figure 7. Comparing this to Figure 5, we see that revenue increases when BA calls have longer holding times. Thus the potential gain from the IPA algorithms as compared to strict partitioning will be more when the BA holding times are longer. Further, since BA arrivals and departures are less frequent when the arrival and departure rates decrease simultaneously, we expect the computational complexity of IPA to decrease due to the fact that IR calls will “see” fewer peaks in the BA profile.

Since the average BA holding time is ten times the average holding time for IR calls, it is reasonable to expect the *near decomposability* property explained in Section 4 to hold. A comparison of the revenue in Figure 7 to the results in the nearly-decomposable case presented in Table 1 shows that the simulation results are close to those predicted by the Markov chain computations. As in Table 1, the revenue is relatively insensitive to r since the traffic intensity of BA calls is small relative to the capacity of the link.

In all our examples, the revenue computed assuming a nearly-decomposable structure was an upper bound on the revenue achievable through any of our admission control schemes for the actual model. We conjecture that this property holds more generally, so that the analytic results in Section 4 should

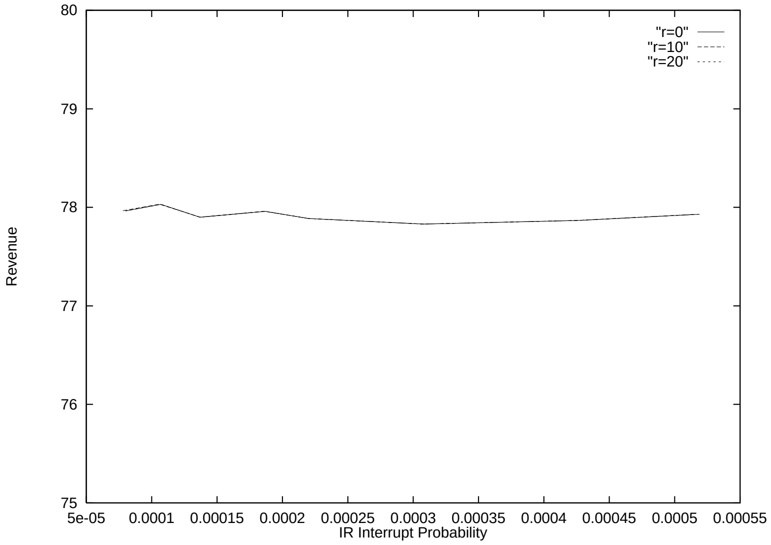

Figure 7: Example 4: p_I versus R for $\mu_I/\mu_B = 10$

serve as a useful reference point.

7. Non-Exponential IR Holding-Time Distributions

On dropping the exponential holding-time assumption for IR calls, we lose the memoryless property of the IR holding times. This makes the computation of the interrupt probability complex. Given the holding-time cdf G and the elapsed holding time x for any call in progress, the remaining holding time of this call has complementary cdf

$$H_x^c(x) \equiv 1 - H_x(y) = G^c(x+y)/G^c(x) \ , \quad y \geq 0 \ , \tag{7.1}$$

where $G^c(x) = 1 - G(x)$. Hence, given the elapsed holding times $x_1, \ldots, x_n$ for n calls in progress, the remaining holding times are independent random variables with cdf's $H_{x_1}, \ldots, H_{x_n}$, defined as in (7.1). Since these cdf's are different, the exact computation of future events is a difficult combinatorial problem.

Fortunately, the following simplification appears to be quite effective. We ignore the elapsed holding times, which reduces the amount of information that we need to store. Ignoring the elapsed holding times, we assume a Poisson arrival process and make an infinite-server approximation. Then, conditioned on there being n calls in progress at some time in equilibrium, the remaining holding times of these n calls are distributed as independent random variables with cdf G_e, which is the stationary-excess cdf given by

$$G_e(t) = \mu_I \int_0^t G^c(u)du.$$

This property holds because the arrival-time and holding-time pairs are distributed according to a Poisson random measure on the plane. We give additional details in the Appendix. An alternative proof of this result can be found in [6].

The infinite-server approximation ignoring elapsed holding times makes it possible to directly extend the FTA, OPA and IPA admission control algorithms to non-exponential holding-time distributions, without increasing the computational complexity. For example, instead of (5.4), the interrupt

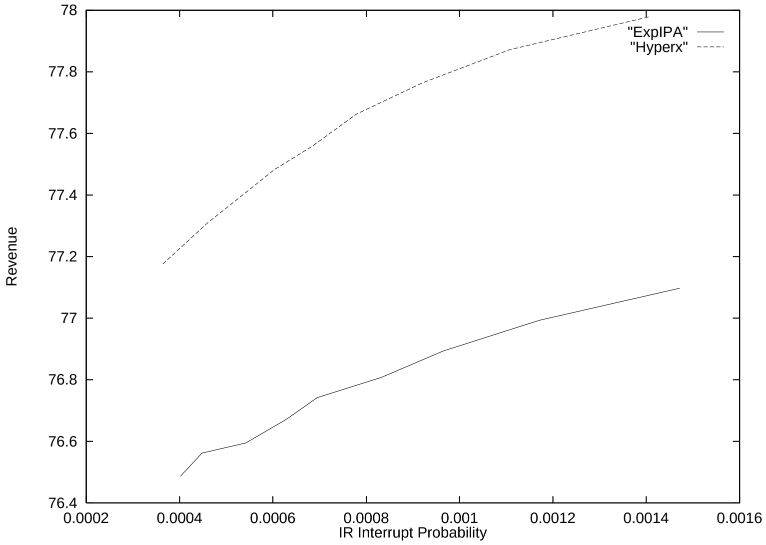

Figure 8: Example 5: p_I versus R for ExpIPA and Hyperx.

probability for IPA becomes

$$p(T_0) = \sum_{i=1}^{\ell} \sum_{k=0}^{N(T_0)+(i-1)b-1} \binom{I(T_0)}{k} \Big(G_e(T_i - T_0)\Big)^k \Big(1 - G_e(T_i - T_0)\Big)^{I(T_0)-k} \Big(1 - G(T_i - T_0)\Big) \ ; \quad (7.2)$$

i.e., the old calls have cdf G_e, while the new call has cdf G. Expressions for the OPA and FTA algorithms can be obtained in a similar fashion.

Example 5: To illustrate we consider the parameters of Example 3 with $r = 50$ and let the holding-time distribution be hyperexponential with balanced means, overall mean $1/\mu_I$ squared coefficient of variation c^2. With the hyperexponential distribution, when a call arrives, with probability q, it chooses a holding time from an exponential distribution with mean $1/\mu_1$ and, with probability $1-q$, it chooses a holding time from another exponential distribution with mean $1/\mu_2$. This model is natural to represent two subclasses of IR calls with different exponential holding times. The balanced means assumption specifies one parameter by requiring that $q/\mu_1 = (1-q)/\mu_2$. Each simulation run was for $500,000$ time units, after deleting 25 time units to get rid of transients. Each curve is plotted based on 10 simulation runs. The simulation runs are longer than in Example 2 to get suitably small confidence intervals and reasonably smooth curves. This is due to the increased variability of the holding-time distributions [20].

We compared IPA based on the hyperexponential distribution (denoted by Hyperx) with IPA based on the exponential distribution (denoted by ExpIPA); i.e., ExpIPA uses the IPA algorithm in Section 5.2 assuming the exponential distribution when the actual holding times are hyperexponential. The purpose of this comparison is study the performance of Hyperx as well as to check whether or not the ExpIPA is sensitive to holding-time distributions. For $c^2 = 10$, Hyperx and ExpIPA are compared in Figure 8. The performance of Hyperx is clearly superior to that of ExpIPA. This example shows that knowledge of the holding-time distribution of IR calls can be exploited to improve the revenue without sacrificing computational complexity.

Example 6: We now consider the case in which the holding times of IR calls are deterministic. We consider this case because we can compare the results to CHTA which is clearly optimal for a fixed value of r. Therefore, consider the same parameters as in Example 5 with the only difference being that the holding times of IR calls are deterministic. Note that the $G_e(t) = t\mu_I$ for $t \leq T$ and is equal to 1 for $t > T$. Each simulation run was for $100,000$ time units, after deleting 25 time units to get rid

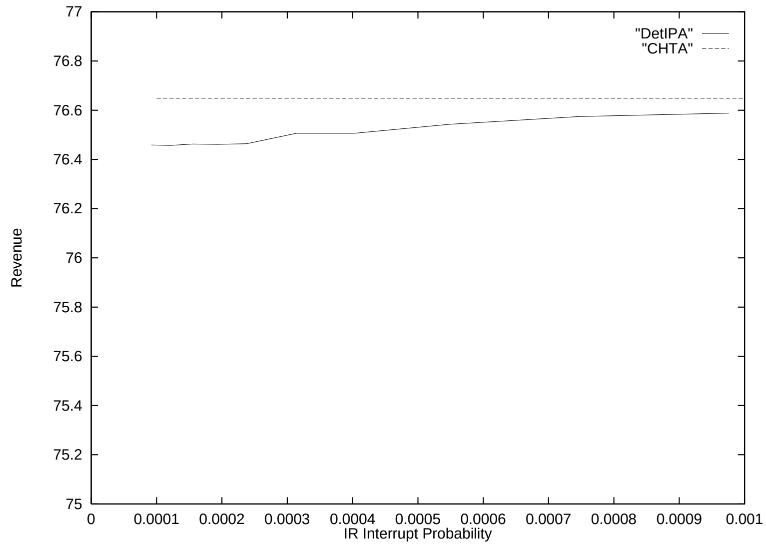

Figure 9: Example 6: p_I versus R for ExpIPA and DetIPA.

of transients. Each curve is plotted based on 10 simulation runs. DetIPA and CHTA are compared in Figure 9. In Figure 9, the CHTA curve is the constant, allowing no interruptions. The gap between the CHTA and DetIPA is what is lost by not keeping track of and exploiting the ages. However, the curves in Figure 9 indicate that this gap is small. Thus, the infinite-server approximation is indeed good and is nearly optimal in the only case for which we know the optimal solution.

8. When BA Calls Need Not Book Far Ahead

So far, we have required BA calls to book far head relative to IR holding times. That clearly is an important case, but it is also of interest to consider what happens when that assumption is relaxed, which we now do.

First, when BA calls do not necessarily book far ahead, the admission control policy needs to be revised. We now might elect to interrupt BA calls. We might also elect to block BA calls if they adversely affect IR calls. Hence, in this more general situation we propose using an interrupt probability calculation for BA calls too.

A BA call is admitted (scheduled in the future) if

(1) there is room for it considering only previously booked BA calls, and

(2) if the interrupt probability will not exceed another the interrupt probability threshold (possibly, but not necessarily, the same threshold as applied to IR calls) after this call is admitted.

It is now less clear which call should be interrupted when there is contention. There are two natural policies: First, we can interrupt the call that arrived most recently, whatever its type. (By the arrival time of a BA call, we mean the time it made its request, not the time it starts service.) Second, we can interrupt the IR call that arrived most recently, and thus never interrupt a BA call. The second policy would be preferred when BA calls are regarded as much more important or valuable. On the other hand, it might be thought that admitted BA calls only ought to be given priority if they book far ahead. Our algorithms can be used with these interruption policies, as well as others.

As before, it may be desirable to provide additional service guarantees, such as trunk reservation, but now these guarantees could be applied to either class. As in [3], we could assign upper limits to both classes. Alternatively, we could apply trunk reservation to one of the classes. With trunk reservation, we propose taking the reservation parameter into account in the interrupt probability computation.

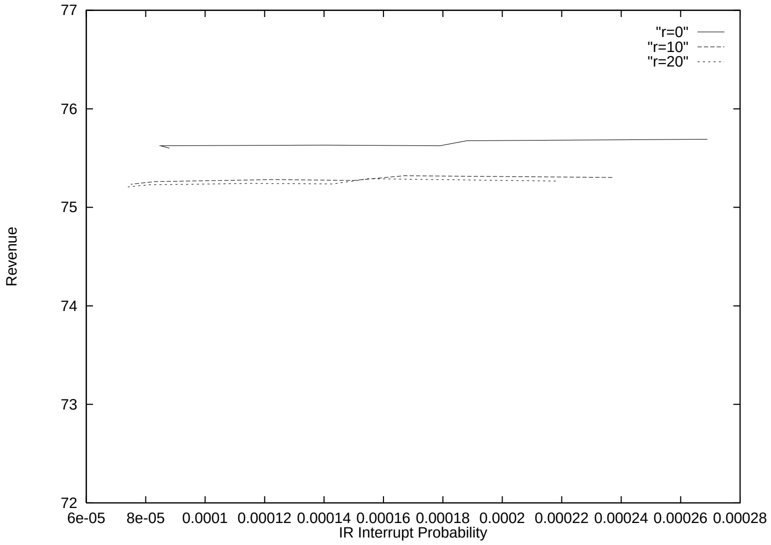

Figure 10: Example 7: p_I versus R

Specifically, if there is a reservation r against BA calls, and a BA call is under consideration, then we compute the probability that the demand will ever exceed $s-r$, assuming that the BA call is admitted. If this calculated probability exceeds a specified threshold, then the BA call is blocked. Similarly, if there is a reservation r against IR calls and an IR call is under consideration, then we consider the probability that the demand will ever exceed $s-r$, assuming that the IR call is admitted. If this calculated probability exceeds the specified threshold, then the IR call is blocked. Of course, we apply trunk reservation against only one of the two classes.

Example 7: To consider the case in which BA calls need not book far ahead, we first consider an example in which the BA book-ahead time is random, allowing any value greater than 0. We consider the same parameters as in Example 3 except that we assume the book-ahead time is hyper-exponentially distributed with mean 20 and squared coefficient of variation (SCV) 5. (The squared coefficient of variation is the ratio of the variance to the square of the mean.) The density of a hyperexponential distribution is decreasing, so that shorter values are most likely, but nevertheless the mean is quite large compared to IR holding times, which we still assume have mean 1. Assuming that hyper-exponential distribution has balanced means, this implies that the book-ahead times are chosen from a a mixture of two exponential distributions with means 11.01 and 109.0 (see [20, page 36, Example 10.2] for the necessary calculations.) Figure 10 plots the IR interrupt probability p_I versus revenue $\mathcal{R}$ for three different values of r. Unlike in the previous examples, since some BA calls do not book very far ahead, p_B may not be close to zero. However, for the value of r which gives the best revenue (see the curve $r=0$ in Figure 10) the BA interrupt probabilities were less than 7×10^{-5} which is clearly very small. Thus, our admission control algorithm still yields better revenue than strict link partitioning when the average BA time is large but there is significant variability in the BA time. However, as it is to be expected, the performance deteriorates when the BA time is highly variable (compare Figure 10 with the results for Example 3).

For the case $r=0$, for the range of IR interrupt probabilities shown in Figure 10 was obtained by choosing the interrupt probability threshold P in the interval $[0.003, 0.03]$. For this range of P, the IR blocking probability was nearly constant around 0.059 and the BA blocking probability varied from 0.03 to 0.045. This shows that, if the average book-ahead time is large, even when there is high variability in the book-ahead times, we can get more revenue with our admission control compared to strict link partitioning and also keep the blocking probabilities of the two call types reasonably small.

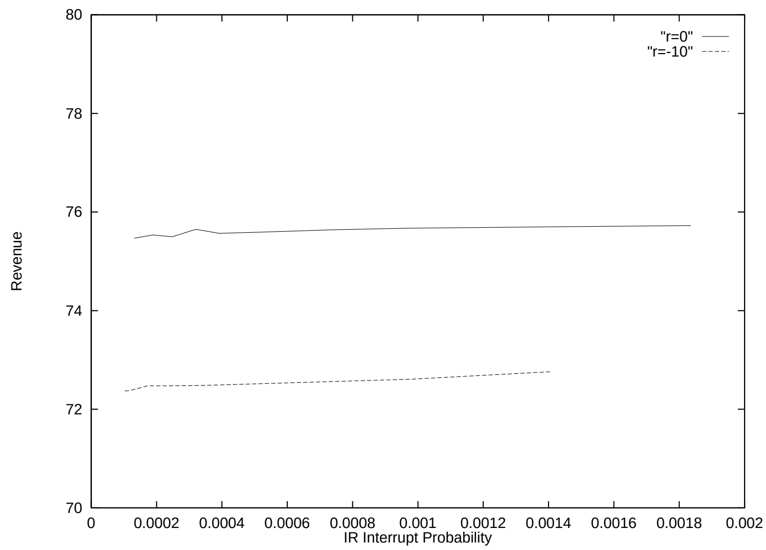

Figure 11: Example 8: p_I versus R

Example 8: We conclude this section with an example which illustrates when strict link partitioning may be preferable to our admission control. Consider the parameters of Example 7 except that the BA time is now assumed to be exponentially distributed with mean 1. Thus, the average BA time is the same as the average holding time of IR calls. In general, if the BA calls are typically conference calls, we expect the BA times to be much larger. But for the sake of completeness, we consider this scenario.

Figure 11 plots the IR interrupt probability p_I versus revenue $\mathcal{R}$ for two different values of r. In both cases p_B was always less than 10^{-4}. From Figure 11, for the case $r = 0$, the revenue using the IPA algorithm is superior to link partitioning (see Table 2). The range of IR interrupt probabilities shown in this figure were obtained by choosing the interrupt probability threshold P in the interval $[0.03, 0.3]$. For this range of P, the IR blocking probability P_I was nearly constant at 0.03 and the BA interrupt probability P_B varied from 0.10 to 0.12. Hence, P_B is much larger than P_I in this case. Thus, if the goal is not just maximum revenue but also to lower the high blocking probability of large bandwidth calls through book ahead, it is natural to reserve some space for BA calls. We do this by choosing $r = -10$, and choosing the same set of values for P. (Recall that a negative value of r indicates that the reservation is against IR calls.) For $r = -10$, P_B varied from 0.094 to 0.098 and P_I from 0.066 to 0.082. Now P_B is less and P_I is more compared with $r = 0$. However, P_I, P_B and $\mathcal{R}$ are all roughly equal to the corresponding values with strict link partitioning and $s_B = 40$ in Table 2. This shows that, when book-ahead times are short, strict partitioning might perform just as well as an admission control algorithm based on calculating interrupt probabilities.

9. Conclusions

In this paper we have proposed an admission control algorithm to use when there are both book-ahead (BA) calls (with specified book-ahead and holding times) and instantaneous- request (IR) calls (with unspecified holding times). We have considered the case of a single link, but the analysis extends directly to networks, assuming fixed routing. To be admitted, a call must satisfy the specified conditions on all required resources. We assume the IR call holding times all have a known common distribution, which may be exponential (Sections 5 and 6) or arbitrary (Section 7). Our main idea is to allow occasional service interruption or service degradation. Our admission control policy is based on determining, under the assumption that the new call is admitted, whether or not the probability a call in progress will eventually need to be interrupted (or have service degraded) exceeds a specified threshold. The new arrival is admitted if the interruption probability is below the threshold; otherwise the call is blocked.

Effective real-time control is achieved by efficiently calculating an approximate value for the interrupt probability. Several approximation schemes were proposed. Simulation experiments showed that the independent-peaks approximation (IPA) yielded better performance than the other approximations and, at the same time, produces a feasible computation for real-time control.

Overall, from extensive simulation experiments we draw the following conclusions:

1. Allowing occasional service interruptions or degradation of service can yield greater revenue than admission control schemes which do not allow them.

2. The IPA scheme can significantly outperform the other candidate approximation schemes for calculating the interrupt probability.

3. The addition of an upper limit or trunk reservation control on BA calls provides more flexibility to achieve a desired balance between BA and IR performance, and can yield additional net revenue.

4. The general-holding-time-distribution algorithm in Section 7 can outperform the exponential-holding-time-distribution algorithm in Section 5 if the holding-time distribution is not nearly exponential (Section 7).

5. The nearly-decomposable Markov chain (ND-MC) algorithm in Section 4 provides a useful approximation when BA calls book far ahead and have relatively long holding times.

6. More generally, revenue tends to increase when the BA time scale (mean holding and interarrival times) increases, so that the limiting case described by the ND-MC algorithm tends to provide an upper bound on achievable revenue, and so is a useful theoretical frame of reference, along with well-known algorithms for the case in which no calls book ahead.

7. Performance can degrade if BA calls do not book relatively far ahead (Section 8).

References

[1] ACCUNET Bandwidth Manager. http://www.att.com/data/data_net/abm.html.

[2] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, Englewood Cliffs, NJ, 1987.

[3] G. L. Choudhury, K. K. Leung and W. Whitt. Efficiently providing multiple grades of service with protection against overloads in shared resources. *AT&T Technical Journal*, 74: 50–63, 1995.

[4] P. J. Courtois. *Decomposability – Queueing and Computer System Applications.* Academic Press, London, 1977.

[5] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes.* Springer-Verlag, New York, NY, 1988.

[6] A. Das and R. Srikant. Diffusion approximations for models of congestion control is high-speed networks. To appear in the *Proceedings of the 37th IEEE Conference on Decision and Control*, Dec. 1998. Longer version of the paper in http://tesla.csl.uiuc.edu/~srikant/pub.html.

[7] M. Degermark, T. Kohler, S. Pink and O. Schelen. Advance reservations for predictive service. In *NOSSDAV '95*, 1995.

[8] Z. Dziong and L. Mason. Control of multi-service loss networks. In *Proc. of the 28th IEEE Conference on Decision and Control*, December 1989.

[9] S. G. Eick, W. A. Massey and W. Whitt. The physics of the $M_t/G/\infty$ queue. *Operations Research*, 41: 731–742, 1993.

[10] D. Ferrari, A. Gupta and G. Ventre. Distributed advance reservation of real-time connections. In *NOSSDAV '95*, 1995.

[11] J. Y. Hui. Resource allocation for broadband networks. *IEEE J. Sel. Areas Commun.*, 6: 1598–1608, 1988.

[12] J. S. Kaufman, Blocking in a shared resource environment, *IEEE Trans. Commun.* COM 29 (1981), 1474–1481.

[13] F. P. Kelly, Notes on effective bandwidths, *Stochastic Networks*, F. P. Kelly, S. Zachary and I. Ziedins (eds.), Clarendon Press, Oxford, 1996, pp. 141–168.

[14] M. Naghshineh and M. Schwartz. Distributed call admission control in mobile/wireless networks. *IEEE J. Sel. Areas Commun.*, 14: 711-717, 1996.

[15] R. G. Phillips and P. V. Kokotovic. A singular perturbation approach to modeling and control of Markov chains. *IEEE Trans. Automatic Control.*, AC-26: 1087-1094, 1981.

[16] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*, Springer-Verlag, 1985.

[17] J. H. Reif and S. Sen. Randomized algorithms for binary search and load balancing on fixed connection networks with geometric applications, *1991 ACM Symposium on Parallel Algorithms and Architectures*, 327–337, Crete, Greece, July 1991.

[18] J. W. Roberts, A service system with heterogeneous user requirements, in *Performance of Data Communication Systems and Their Applications*, G. Pujolle (ed.), North Holland, Amsterdam, 1981, pp. 423–431.

[19] K. W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer, London, UK, 1995.

[20] R. Srikant and W. Whitt. Simulation run lengths to estimate blocking probabilities in multiserver loss models. *ACM Transactions on Modelling and Computer Simulation*, 6: 7–52, 1996.

[21] R. Srikant and W. Whitt. Resource sharing for book-ahead and instantaneous-request calls using a CLT approximation. Submitted to *Telecommunication Systems*, 1998.

[22] D. Wischik and A. Greenberg, Admission control for booking ahead shared resources, *Proceedings IEEE Infocom '98*, 1998.

[23] L. C. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller and H. Wittig. Issues of reserving resources in advance. In *NOSSDAV '95*, 1995.

Appendix.

Consider an $M/GI/\infty$ system with arrival rate λ_I and holding-time cdf G with mean $1/\mu_I$. Let s denote the arrival time and u denote the call's holding time of an arbitrary call. It was pointed out in [9] that the set of all such pairs (s, u) is a generalized Poisson process or a Poisson random measure [5, Chapter 2.4]. Thus, if $N(A)$ denotes the number of arrival time-holding time pairs in A, where A is a subset of $\mathbf{R}^2$, then

$$P(N(A) = n) = \frac{e^{-\Lambda(A)}(\Lambda(A))^n}{n!}, \tag{A.1}$$

where

$$\Lambda(A) \equiv \lambda_I \int\int_A dG(u)ds.$$

Moreover, if k is any integer with $k \geq 2$ and $A_1, \ldots, A_k$ are disjoint subsets of $\mathbf{R}^2$, then $N(A_1), \ldots, N(A_k)$ are independent random variables.

The Poisson random measure property critically depends on the arrival process being Poisson and the number of servers being infinite. We use this property to show that, conditioned on there being n calls in progress at some time in steady-state, the remaining holding times of these n calls are distributed as independent random variables with cdf G_e. The proof below exploits the properties of Poisson random measures along the lines of the proofs of Theorems 1 and 2 of [9].

For a set A, let $|A|$ be its cardinality (the number of elements). Let $N(t)$ denote the number in the system at time t, i.e.,

$$N(t) = |\{(s, u) | s + u \geq t, s \leq t\}|,$$

and $M(t, t_1, t_2)$ denote the number in the system at time t with remaining holding times in the interval $[t_1, t_2)$, i.e.,

$$M(t, t_1, t_2) = |\{(s, u) | s + u \in [t + t_1, t + t_2), s \leq t\}|.$$

Proposition 1. *As $t \to \infty$, $M(t, t_1, t_2)$ converges in distribution to a Poisson random variable with mean $\frac{\lambda_I}{\mu_I}(G_e(t_2) - G_e(t_1))$.*

Proof: By the Poisson random measure property, $M(t, t_1, t_2)$ has a Poisson distribution for each t. From the definition of $M(t, t_1, t_2)$, its mean is given by

$$\int_0^t \int_{t+t_1-s}^{t+t_2-s} \lambda_I dG(u)ds = \int_0^t \lambda_I(G^c(t + t_1 - s) - G^c(t + t_2 - s))ds.$$

Defining $r_1 \equiv t + t_1 - s$ and $r_2 \equiv t + t_2 - s$, the above expression becomes

$$\lambda_I \int_{t_1}^{t+t_1} G^c(r_1)dr_1 - \lambda_I \int_{t_2}^{t+t_2} G^c(r_2)dr_2,$$

which in the limit as $t \to \infty$ is given by

$$\frac{\lambda_I}{\mu_I}(G_e^c(t_1) - G_e^c(t_2)) = \frac{\lambda_I}{\mu_I}(G_e(t_2) - G_e(t_1)).$$

$\square$

Given that there are n calls in the system, let us index them arbitrarily from the set $\{1, 2, \ldots, n\}$. Let T_{e_i} denote the remaining holding time of call i.

Proposition 2.

$$\lim_{n\to\infty} P(T_{e_1} \leq t_1, T_{e_2} \leq t_2, \ldots, T_{e_n} \leq t_n \mid N(t) = n) = G_e(t_1)G_e(t_2)\ldots G_e(t_n).$$

Proof: Assume that $t_i \neq t_j$ for all i and j.

$$\lim_{t\to\infty} P(T_{e_i} \in [t_i, t_i + dt_i)\ \forall i \in \{1,2,\ldots,n\} | N(t) = n) \tag{A.2}$$

$$= \lim_{t\to\infty} \frac{P(T_{e_i} \in [t_i, t_i + dt_i)\ \forall i \in \{1,2,\ldots,n\}, N(t) = n)}{P(N(t) = n)} \tag{A.3}$$

$$= \lim_{t\to\infty} \frac{\frac{1}{n!} P(M(t, t_{i-1}, t_i) = 0, M(t, t_i, t_i + dt_i) = 1\ \forall i \in \{1,2,\ldots,n\}, M(t, t_n, \infty) = 0)}{P(N(t) = n)} \tag{A.4}$$

$$= \lim_{t\to\infty} \frac{\frac{1}{n!}\Big[\Pi_{i=1}^{n} P(M(t, t_{i-1}, t_i) = 0) P(M(t, t_i, t_i + dt_i) = 1)\Big] P(M(t, t_n, \infty) = 0)}{P(N(t) = n)} \tag{A.5}$$

$$= \frac{\frac{1}{n!}\Big[\Pi_{i=1}^{n} e^{-\frac{\lambda_I}{\mu_I}(G_e(t_i) - G_e(t_{i-1}))} \frac{\lambda_I}{\mu_I} dG_e(t_i)\Big] e^{-\frac{\lambda_I}{\mu_I}(1 - G_e(t_n))}}{\frac{1}{n!} e^{-\frac{\lambda_I}{\mu_I}} \left(\frac{\lambda_I}{\mu_I}\right)^n} \tag{A.6}$$

$$= dG_e(t_1) dG_e(t_2) \ldots dG_e(t_n), \tag{A.7}$$

where $t_0 = 0$. In the above set of equations, (A.4) follows from the definition of $M(t, t_1, t_2)$ and the $n!$ possible orderings of $(T_{e_1}, \ldots, T_{e_n})$, (A.5) follows from the independence of counts in disjoint subsets and (A.6) follows from (A.1) and *Proposition 1*. Finally, the general case in which $t_i = t_j$ for some i and j follows from the right continuity of the joint cdf.