A preemptive connection pool manager for web-based application collaboration

Zhen Zhao
Comcast, 1701 John F. Kennedy Blvd, Philadelphia, PA 19103
email: zhen_zhao@comcast.com

***Abstract*—Most of web-based applications are free marketing systems. They compete network resources usually in first-come-first-service (FCFS) mechanism. As a big company with thousands of applications, Comcast faces how to maximize its revenue through all these applications. Currently, all these applications have no collaborations even they are considered in one big system. When the resource limit is reached and increasing the capacity is not an applicable method, FCFS may degrade critical services with running some less important applications. We propose a priority-based collaboration solution. Every application has its pre-determined priority. When the network resource is all occupied, new incoming requests from applications with higher priorities preempt those with lower priorities and so the high priority work can be processed with sacrificing the low priority work's performance. Specifically, we implement a connection pool manager that admits new connections for the critical requests at the cost of preempting out some less important connections when the system safety limitation is approached. The major win here is that we increase our system usability for the most important features and maximize the revenue during a traffic burst exceeding our resource capacity.**

***Index Terms*—loss network, admission control, preemption, coordinate convex, dynamic programming**

I. Introduction

Applications may be divided into two categories: i) first-come-first-service (FCFS) applications and ii) priority-based applications, see Fig. 1. The current Internet is an example of a FCFS network and so are most of web-based applications. It is well-known that FCFS is appropriate for the traditional data service but do not guarantee the adequate performance of critical applications. Priority-based applications are introduced to guarantee higher priority service gains higher quality of service (QoS). A well-known mechanism is admission control. However, it is also known as inefficient for bursty connections. Preemption is a method allowing FCFS when there is still free resource and starting priority-based control when the remaining resource is not enough to handle the new incoming request. The priority-based control of preemption is dropping the work of lower priority to free some resource for the incoming higher priority requests.

In recent years, collaborations between human beings are becoming more and more important. However, few efforts are focused on the collaboration of web-based applications. Most current web-based applications are running on networks as in a free market. Even within an organization or a website, FCFS is applied by most web-based applications. There are no collaborations between them.

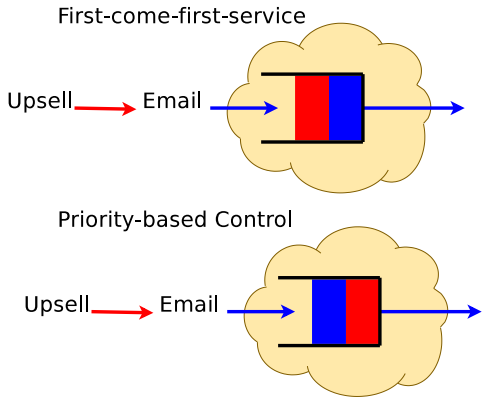

Fig. 1. A single queue model showing First-come-first-service (FCFS) does not control resources while priority-based control rearrange the "Upsell" job ahead of "email" even "email" request arrives first.

Web-based applications within one organization usually share certain http connection pools. Http requests comprise a rapidly growing fraction of the connections from client side to service in recent years. No collaborations between these requests may result in blocking critical requests when the connection pool is full. Due to the limitation of resource and the bursty features of http requests, avoiding connection pool being filled up is not always feasible. Therefore, a mechanism to better take care of this scenario becomes necessary.

Some web-based applications such as "upsell" is fundamentally different from other elastic applications (e.g., data transfer applications like web and email) in terms of their direct contributions to the revenue. As a profit company, satisfactory performance of these critical applications requires quality of service (QoS) guaranty such that these requests are serviced even the connection pool is full. Thus, a collaboration mechanism is necessary to rearrange the connection resource.

Kelly [1] presents a model that is defined as a collection of connections, where a control mechanism determines whether or not to admit each arriving call on each connection.

The importance of certain web-based applications to an organization and the fact that Kelly's model is the appropriate architecture to offer the QoS guarantees that such applications requires motivates this study of applying loss networks control mechanism into web-based.

Http connection pools offering multiple service classes are capable of discriminating among different connection requests,

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250465

see Fig. 2. Multi-class connection pools service multiple classes of calls, where classes often indicate call priority, and call priority often reflects the importance to the organization for each admitted call. In the general case arrival rate, service rate, and call rate/size (the number of circuits on each connection consumed by a call of that class) are class specific. The importance of multi-class service discrimination arises from the widely heterogeneous nature of collaborated web-based applications, ranging from casual email recommendations to critical services (e.g., upsell). This importance motivates our study of multi-class connection pools.

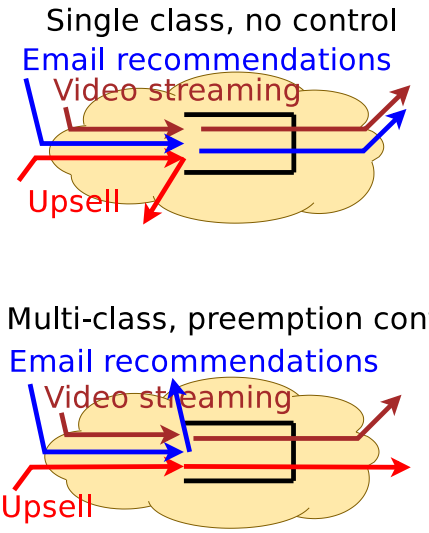

Fig. 2. Multi-class connection pools are capable of service discrimination whereas single class connection pools are not.

The facts that connection pools have a finite amount of circuits and that admitted calls reserve resources requires that connection managements employ some form of control to limit resource consumption. There are three popular control mechanisms for multi-class connections control: admission control, preemption control, and capacity adaptation. See Fig. 3. The most widely used control mechanism in is admission control. An admission control policy specifies whether or not to admit an arriving call of a given class as a function of the number of active calls of each class in the pool.

A second control mechanism for connection pool is capacity adaptation, where incoming calls may ask for dynamically adjusting the connection pool capacity in response to changes in the instantaneous connection occupancy. How the capacity of a pool changes and by how much, are specified by the adaptation policy.

A third control mechanism for multi-class connection pools is preemption, where an arriving call may be admitted by possibly preempting an active connection of lower priority. The preemption policy is typically a function of the number of active calls of each priority level, which we call the state of the pool. The preemption policy specifies whether to i) block, ii) admit without preemption, or iii) admit with preemption an arriving call of each possible class as a function of the

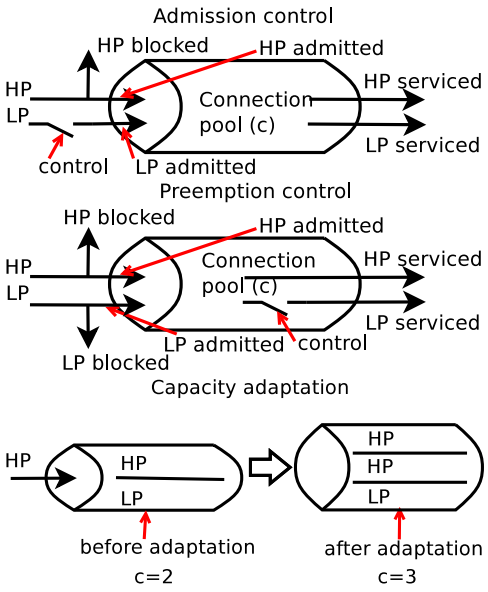

Fig. 3. Control mechanisms for multi-class connection pools include admission control, preemption and adaptation.

state. The preemption policy enables service differentiation in that the blocking probability is typically smaller for higher priority calls, but incurs the cost that lower priority calls may find themselves admitted then preempted before their requests get intended responses. Preempted calls may be rerouted (for example through a proxy) or put into the backlog of the queue or dropped depending upon the network policy and resource availability. In short, preemption may be used to assure that high priority calls are established along its desired connections. The use of preemption policies for critical calls in connection pools has gained attention in recent years as a flexible and effective control mechanism to dynamically allocate resources among competing web service requests with different priorities. This paper addresses the performance analysis and policy design of preemptive multi-class connection pool within Comcast , see Fig. 4.

The contributions of this paper includes: 1) proposing a preemption policy and investigating the performance of a http connection pool servicing multiple service classes under a specified preemption policy. This work is the first to apply preemption to web-based application collaborations. It successfully analyzes the preemption rates under preemption and characterizes the preemption rates/probabilities for each of K preemptive classes with homogeneous service rates and the limitation of those with heterogeneous service rates. The proof of the proposed preemption policy being independent of the admission optimization allows us to use this policy in addition to the known optimization strategies.

This paper is organized as follows: Section II presents related work to this work. Section III presents a simple 2-class homogeneous service rate model and performance analysis of the proposed preemption policy on the connection pool. An extension from 2-class to generic K-class is presented in

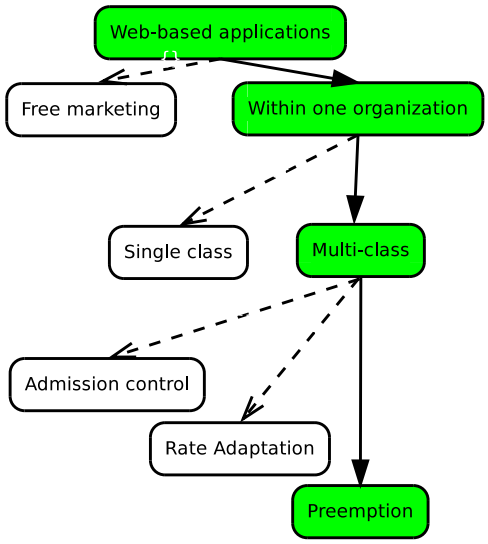

Fig. 4. Preemption control of multi-class loss networks and related context.

Section IV. Section V discussed the extension limitations of heterogeneous service rates. Section VI proves the proposed preemption policy fit the known optimal admission control. Simulation results are shown in Section VII.

II. Related Work

We divide our discussion of related work on preemption into two parts. The first part discusses related work on proposed preemption policies, both optimal and heuristic. Although much of this work discusses the important issue of the computational complexity of the proposed policies, in general this body of work contains very little in the way of performance analysis. The second part concentrates on performance analysis of a model with preemptive priority.

A. Proposed preemption policies

The 1992 paper by Garay and Gopal addressed the call preemption problem in communication networks [2], showing that the problem of selecting a connection for preemption in order to minimize the number of preempted connections or minimize the amount of preempted bandwidth is NP-complete. They propose heuristics for a centralized network framework that are shown to perform reasonably well relative to the optimal solution. Extending Garay and Gopal's work, in 1997 Peyravian and Kshemkalyani proposed decentralized network connection preemption algorithms [3] that optimize three fixed criteria in a given order of importance: number of connections, bandwidth, and priority.

After these two seminal works, many of the subsequent proposed preemption policies have been described in the context of a Differentiated Services (DiffServ) aware MPLS scenario, *e.g.*, [4], [5], [6], [7], [8], [9], [10], discussed below. In particular, the decentralized policies in [3] are the basis for our earlier work on flexible and adaptive preemption policies [4]. Here, an order of importance for the considered criteria is not fixed, but can be configured by the network provider according to the network's best interest. In [5], Sung-eok *et al.* propose a centralized connection preemption algorithm that optimizes the preemption criteria in a fixed order different from [3]. In [6], Tong *et al.* present an algorithm that jointly considers both bandwidth allocation and preemption.

Stanisic and Devetsikiotis propose simple preemption policies based on random selection; this dramatically reduces the time needed to select a set of connections to be preempted [7]. Both Blanchy *et al.* [8] and Yu *et al.* [9] focus on preemption–aware routing algorithms. In particular, a route is selected by minimizing the number of connections (LSPs) that require preemption. The routing algorithm therefore tries to minimize the occurrence of preemption events and thereby minimize the need for rerouting. Recently, Vieira and Guardieiro implemented de Oliveira's preemption policies in [4] using fuzzy logic and genetic algorithms in an MPLS testbed [10].

B. Analysis of preemption

Ours is the first analytical treatment of the performance of a preemptive connection pool. Related work studies either a *network* servicing multi-class elastic (*e.g.*, email, web) or inelastic (*e.g.*, voice, video) traffic *with preemption*, or a *general network* servicing multi-class traffic (elastic or inelastic) *without preemption*. The text by Ross [11] covers non-preemptive loss networks (for inelastic traffic), while the text by Srikant [12] covers non-preemptive best-effort networks (for elastic traffic). Below, we restrict our attention to work on preemption modeling.

Preemptive systems can be dichotomized into preemption with delay and preemption with loss. Preemption with delay means preempted calls are "put on hold", and queued untill their service resumes or restarts. Preemption with loss means that preempted calls are removed, this can mean either transfer or eviction. Preemption with delay is usually modeled by an *M/G/c* queue (infinite queueing), while preemption with loss is usually modeled by an *M/G/c/c* queue (no queueing).

Preemption with delay. The earliest analysis of preemption is in the context of preemption with delay. In fact, the first paper published on priority queueing with preemption is from 1958, by White and Christie [13]. In this paper, White and Christie analyze the average queue length and the average time in system for a preemptive resume and repeat policy. They also study a "breakdown" system where the preemptive server is prone to fapoolure (vacations). Miller [14] uses matrix-geometric methods to compute steady state probabilities for an *M/M/1* priority queue, modeling a pool servicing elastic traffic with preemption. Buzen and Bondi [15] published an article in 1983 studying a network of *M/M/c* queues with preemptive resume policies. Their results are focused on moments in a preemptive–delay network. Ngo and Lee published a short note in 1990 [16] on a single *M/M/c* queue with preemptive priority, extending [14]. The work in [14] is further generalized by Cho and Un [17], who provide an analysis of a combined preemptive/nonpreemptive priority *M/G/1* queue. There are many other papers in the queueing literature on preemption with delay;

these analyses are of limited relevance to our work since our focus is on preemption with loss.

Preemption with loss. The above articles analyze the performance of a preemption system with delay. Unfortunately, the more prevalent use of preemption policies (*e.g.*, MPLS) is to drop (as in the loss model), rather than postpone (as in the delay model) the preempted calls. There is some existing work on preemption with loss, but all such work is either analysis of a pool, or has a numerical/computational focus for multiple parallel pools. The earliest performance analyses of a preemption policy in a loss context are by Helly [18] and Burke [19], both from 1962. These short papers present the framework for employing the Erlang B blocking probability equation on a pool with preemption. These two papers served as an inspiration for our results in §III. After that, the literature appears to be spilent untill 1980 when Calabrese *et al.* [20] published an analysis of a voice network of multiple parallel pools with preemption. Their paper includes a discussion of a variety of different preemption policies, which they term "ruthless" and "friendly." This model combines the two preemption policies with the *estimated* probability that a high priority call returns to the original pool after searching all alternate pools and finding them blocked. Although this paper studies multiple parallel pools, the focus is on algorithms for computation of the performance metrics, along with numerical approximations of the optimal solution. In contrast, our work focuses on closed-form performance expressions. Moreover, [20] is essentially a "soft" preemption model, where high-priority calls only preempt low-priority calls if each of the routes is full, whereas our "hard" preemption model allows high-priority calls to preempt low-priority calls if the primary pool is full, regardless of the status of the backup pool. In 1980, Fischer [20] discussed the blocking and preemption probabilities of two priority classes with different service times in a single preemptive connection pool. In that paper, due to the difficulty in solving the steady state equations, the author analyzed three special cases of the solution: i) *M/M/1/1*, ii) *M/M/c/c* with ratio of class 2 to class 1 mean holding time tends to 0 and iii) *M/M/c/c* with ratio of class 2 to class 1 mean holding time tends to ∞.

III. Definitions and Modeling Descriptions

In this article, we consider that connections have two priority levels: high (HP) and low (LP). When congestion occurs, preemption is called, and low priority connections are then removed from the pool. We consider two performance metrics: blocking probability and preemption rate, the latter meaning how much of LP connections are preempted from the pool due to the congestion.

Blocking probability: connections are blocked if there is not sufficient capacity to accommodate them at their time of arrival. Preemption is employed then HP connections can free up capacity by preempting LP connections out of the pool, and a HP is only blocked if the only connections in the pool are other HP connections.

Rate of preemption: Number of preemption events per time unit. The time average is calculated as follows:

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbf{1}(\text{preempt at time } t)\, dt \tag{1}$$

where t is the observation duration.

Firstly, we focus on a simple case where a connection pool servicing two classes of connections: HP and LP.

The scenario is depicted graphically in Fig. 5

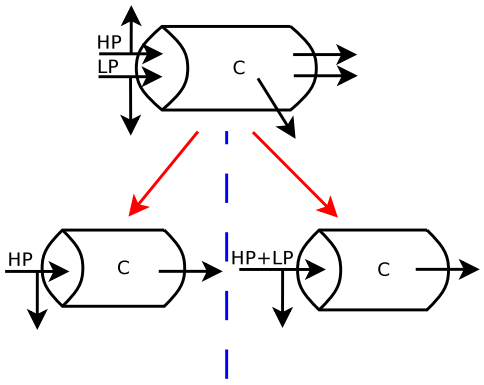

Fig. 5. Preemption control of two-class connection pool.

The high and low priority arrival processes are independent Poisson processes of intensities λ_h and λ_l respectively. Connection durations are exponential with average value $\mu_h^{-1} = \mu_l^{-1} = \mu^{-1}$ seconds. Connections are multiplexed onto the pool of capacity c. Our purpose is to study how our two performance metrics vary in the aggregate offer load $\rho = \lambda_h\mu^{-1} + \lambda_l]\mu^{-1}$ for both high and low priority application requests.

A. Blocking probabilities

When we consider blocking probability of HP connections, it is straightforward that incoming HP connections only "see" other HP connections being serviced under preemption policy. That is, when a HP connection can preempt a LP connection, the blocking of incoming HP connections is unrelated to the state of LP connections. Thus, the HP blocking probability can be computed using the model shown as the left lower subfigure in Fig. 5.

When we consider blocking probability of LP connections, we recognize when a LP connection request arrives, it sees both HP and LP connections. Thus, its blocking events are due to the aggregate occupancy of both HP and LP connections. Therefore, the LP blocking probability can be computer using the model shown as the right lower subfigure in Fig. 5.

Suppose we are not employing preemption. Let n_h, n_l denote the mean number of high and low priority connections at some typical time, i.e., when the pool is in steady state. By Little's Law we can relate n_h, n_l to λ_h, λ_l through

$$n_h = \lambda_h\mu^{-1}, \quad n_l = \lambda_l\mu^{-1}. \tag{2}$$

Grounded on our assumption of the arrival processes of HP and LP are Poisson and service time are exponential, the occupancy of high and low priority connections is modeled as

a $M/M/c/c$ Markov chain, where (n_h, n_l) is the state of this 2-dimensional Markov process.

For an overview of the $M/M/c/c$ queue and the Erlang-B blocking probability equation, the reader is referred to Kelly's [1]. We write $E(\rho, c)$ for the Erlang-B blocking probability of an $M/M/c/c$ queue with arrival rate λ, service rate μ, and offered load $\rho = \lambda/\mu$. For notational convenience we will write $\bar{E}(\rho, c) = 1 - E(\rho, c)$ to denote the admission probability of an $M/M/c/c$ queue.

Extending this idea to finding congestion points when preemption is employed, we find that the congestion arrival rates for high and low priority connections are found in the two lower figures in Fig. 5. Therefore, the blocking probabilities of HP and LP connections are

$$\mathbb{P}(\text{HP blocking}) = E(\rho_h, c),\ \rho_h = \lambda_h \mu^{-1}, \tag{3}$$
$$\mathbb{P}(\text{LP blocking}) = E(\rho_h + \rho_l, c),\ \rho_l = \lambda_l \mu^{-1}. \tag{4}$$

Analysis of admission and service completeness rates for HP connections. Although the HP connections may induce preemption in the LP connections, as we pointed out, the HP connections admission process is in fact independent of the number of LP connections. The pool admission rate A_h and service completeness rate D_h for the HP connections are given by:

$$A_h \quad = \quad D_h = \lambda_h \bar{E}(\rho_h, c).$$

Analysis of admission rates for LP connections. The analysis of LP connections admission rates A_l is siMillar to that of HP connections except that the total load $\rho = \rho_1 + \rho_2$ is used in calculating the blocking probabilities:

$$A_l \quad = \quad \lambda_l \bar{E}(\rho, c).$$

B. Preemption rates

To find out the preemption rates, we need figure out under what conditions preemption policy is employed.

From what we described before, preemption policy is applied when HP connection request sees the pool is filled with HP and LP connections, which is equal to the scenario that LP connections see blocking. In a word, preemption is employed when LP is blocked while HP is not blocked.

The set of states that cause HP blocking, LP blocking, and LP preemption are:

$$\begin{aligned} \mathcal{S}_{\text{HP blocking}} &= \{(c, 0)\}, \\ \mathcal{S}_{\text{LP blocking}} &= \{(n_h, n_l) : n_h + n_l = c\}, \\ \mathcal{S}_{\text{Preemption}} &= \mathcal{S}_{\text{HP blocking}} \setminus \mathcal{S}_{\text{LP blocking}}. \end{aligned}$$

The first equation says a HP connection is blocked at the connection pool iff it arrives to find the pool filled with c HP connections. The second equation says a LP connection is blocked at the pool iff it arrives to find the pool filled with c connections total. The third equation says a HP connection causes a preemption of a LP connection from the pool iff it arrives to find the pool filled with c connections total and one or more of them are LP. See Figure 6 for a picture of these three events.

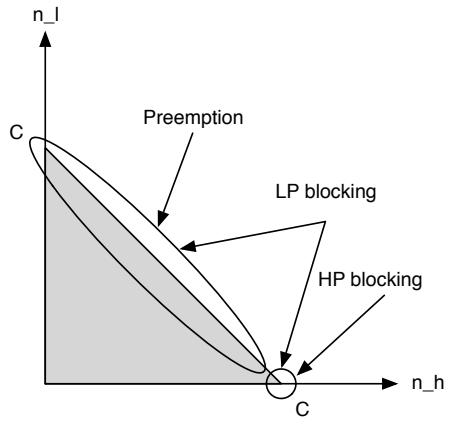

Fig. 6. Illustration of the state space $\mathcal{S}$ for the connection pool with c. The x-axis is the number of HP connections, the y-axis is the number of LP connections.

By PASTA (Poisson arrivals see time averages), the probability of these events are found by summing the invariant distribution over the states comprising the event. The invariant distribution for this system is not known in closed form (to our knowledge) but the probabilities of the events of interest are known. In particular:

$$\begin{aligned} \mathbb{P}(\text{HP blocking}) &= \mathbb{P}((N_h, N_l) \in \mathcal{S}_{\text{HP blocking}}) = E(\rho_h, c), \\ \mathbb{P}(\text{LP blocking}) &= \mathbb{P}((N_h, N_l) \in \mathcal{S}_{\text{LP blocking}}) = E(\rho, c), \\ \mathbb{P}(\text{Preemption}) &= \mathbb{P}((N_h, N_l) \in \mathcal{S}_{\text{Preemption}}) \\ &= \mathbb{P}(\text{LP blocking}) - \mathbb{P}(\text{HP blocking}) \\ &= E(\rho, c) - E(\rho_h, c). \end{aligned}$$

This line of reasoning is originally due to [**?**]. Thus the preemption rate from pool is

$$R = \lambda_h [E(\rho, c) - E(\rho_h, c)]. \tag{5}$$

IV. K-CLASS POOL MODEL

In the last section, we investigated a simple case where 2-class connections with the same service mean duration are put into one pool. In this section, we study how to extend it to a more general case and the limitation of the extension.

A. K-class in a pool

Two direct extensions include: 1) multi-class in the connection pool; 2) the preempted LP connections may be dropped or put into another pool, see Fig. 7.

It is easy to extend the model in the previous section from 2-class to K-class, see Fig. 8 .

By an exactly analogous argument the probability of any request with higher priority than class k connections in a state that would cause a class k or lower priority connection to be preempted from the pool is found to be

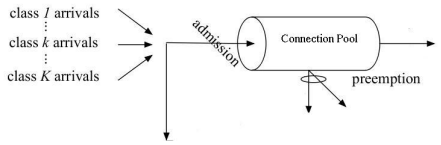

Fig. 7. K-class in the connection pool. Preempted connections may be dropped or put into another pool.

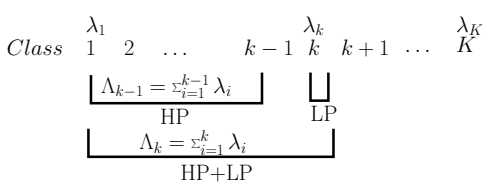

Fig. 8. For K-class connections in a pool, a class k connection request only sees class $1, \ldots, k$ connections in the pool and all the classes $1, \ldots, k-1$ as HP. Therefore, we can group all the $1, \ldots, k-1$ classes as HP and class k as LP for the general cases.

$$\begin{aligned}
\mathbb{P}_k(\text{Preemption}) &= \mathbb{P}\left(\Big(\sum_{i=1}^{k-1} n_i, \sum_{j=k}^{K} n_j\Big) \in \mathcal{S}_{\text{Preemption}}\right) \\
&= \mathbb{P}(\text{class } 1, \ldots, K \text{ blocking}) \\
&\quad -\mathbb{P}(\text{class } 1, \ldots, k-1 \text{ blocking}) \\
&= E\left(\Big(\sum_{i=1}^{K} \lambda_i\Big)\mu^{-1}, c\right) \\
&\quad -E\left(\Big(\sum_{i=1}^{k-1} \lambda_i\Big)\mu^{-1}, c\right).
\end{aligned}$$

Thus the total rate of preemption from the pools caused by class k is

$$R_k = \lambda_k \left[E\left(\Big(\sum_{i=1}^{K} \lambda_i\Big)\mu^{-1}, c\right) - E\left(\Big(\sum_{i=1}^{k} \lambda_i\Big)\mu^{-1}, c\right)\right]. \tag{6}$$

V. Heterogeneous service rates

In this section we discuss the reasons why the heterogeneous service rates case is in general intractable. Heterogeneous service rates mean connections have different service mean durations. This is unrelated to the priority so instead of using h, l to label the two classes, we use $1, 2$ to distinguish them, $\mu_1 \neq \mu_2$. We then discuss an approximate solution valid in a time-scale separation regime.

The primary reason for the intractability of the heterogeneous service rates case is the fact that the CTMC $\{\mathbf{n}(t)\}$ is not lumpable under a partition aligned with the performance metrics of interest.

Occupancy partitions. We introduce two occupancy partitions for the Markov chain $\{\mathbf{n}(t)\}$.

Definition 1: The aggregate occupancy partition *(aop) of* $\mathcal{S}$ *is* $\mathcal{S}_m^{\text{aop}} = \{\mathbf{n} \in \mathcal{S} : N_K = m\}$ *for each* $m = 1, \ldots, c$.

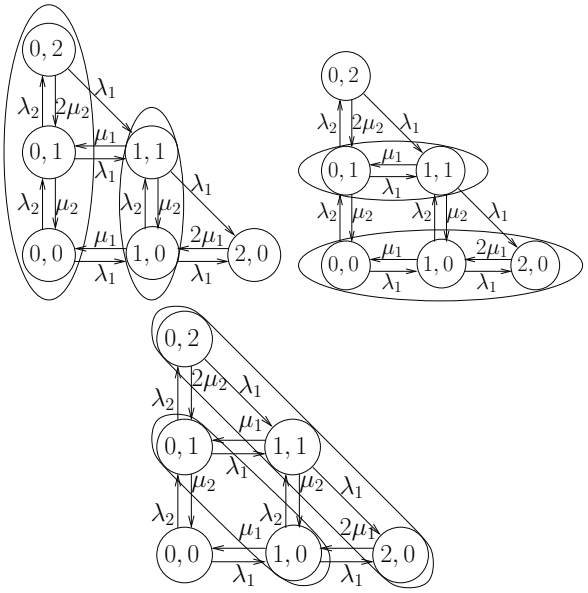

Fig. 9. Illustration of the occupancy partitions for $c = 2$ and $K = 2$. Each state shown represents an occupancy of $\mathbf{n} = (n_1, n_2)$. **Top left:** the priority 1 occupancy partition, **top right:** the priority 2 occupancy partition, **bottom**: the aggregate occupancy partition.

Definition 2: The priority k occupancy partition *(pop-k) of* $\mathcal{S}$ *is* $\mathcal{S}_m^{\text{pop},k} = \{\mathbf{n} \in \mathcal{S} : n_k = m\}$*, for each occupancy level* $m = 0, \ldots, c$ *and some priority level* $k = 1, \ldots, K$.

The *aop*, *pop*-1, and *pop*-2 partitions are shown in Fig. 9 for the case of a pool with capacity $c = 2$ and $K = 2$ priority classes. The following theorem identifies when the Markov chain $\{\mathbf{n}(t)\}$ is lumpable over these partitions.

Theorem 1: The CTMC $\{\mathbf{n}(t)\}$ *is:*

1) *Lumpable under the* aop *with* homogeneous *service rates, and is Markovian across subsets.*
2) Not *lumpable under the* aop *with* heterogeneous *service rates, and therefore not Markovian across subsets.*
3) *Lumpable under the* pop-1 *with homogeneous or heterogeneous service rates, and so Markovian across subsets.*
4) Not *lumpable under the* pop-k *(for* $k > 1$*) with homogeneous or heterogeneous service rates, and therefore not Markovian across subsets.*

Proof: Consider transitions from occupancy level m to $m+1$ and to $m-1$. Let $\mathbf{n} \in \mathcal{S}_m$ be a state in occupancy level m.

1. The transition rate from $\mathbf{n}$ in aggregate occupancy level $m > 0$ to aggregate occupancy level $m-1$ is

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m-1}^{\text{aop}}} q_{\mathbf{n},\mathbf{n}'} = n_1\mu + \cdots + n_K\mu = m\mu, \tag{7}$$

and the transition rate from $\mathbf{n}$ in aggregate occupancy level $m < c$ to aggregate occupancy level $m+1$ is

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m+1}^{\text{aop}}} q_{\mathbf{n},\mathbf{n}'} = \lambda_1 + \cdots + \lambda_K = \Lambda_K. \tag{8}$$

In both cases the transition rate is independent of the state $\mathbf{n}$.

2. The transition rate from $\mathbf{n}$ in aggregate occupancy level $m > 0$ to aggregate occupancy level $m-1$ is:

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m-1}^{\text{aop}}} q_{\mathbf{n},\mathbf{n}'} = n_1\mu_1 + \cdots + n_K\mu_K. \tag{9}$$

The transition rate depends upon the state $\mathbf{n}$.

3. The transition rate from $\mathbf{n}$ in priority 1 occupancy level $m > 0$ to priority 1 occupancy level $m-1$ is:

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m-1}^{\text{pop},1}} q_{\mathbf{n},\mathbf{n}'} = n_1\mu_1 = m\mu_1, \tag{10}$$

and the transition rate from $\mathbf{n}$ in priority 1 occupancy level $m < c$ to priority 1 occupancy level $m+1$ is

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m+1}^{\text{pop},1}} q_{\mathbf{n},\mathbf{n}'} = \lambda_1. \tag{11}$$

In both cases the transition rate is independent of the state $\mathbf{n}$.

4. The transition rate from $\mathbf{n}$ in priority $k > 1$ occupancy level $m > 0$ to priority k occupancy level $m+1$ is:

$$\sum_{\mathbf{n}' \in \mathcal{S}_{m+1}^{\text{pop},k}} q_{\mathbf{n},\mathbf{n}'} = \lambda_k \mathbf{1}_{n_1+\cdots+n_k<c}. \tag{12}$$

The transition rate depends upon the state $\mathbf{n}$.

■

The key reason why the chain is lumpable under the *aop* is that *preemptions do not change the aggregate occupancy level*. It is also worth noting that the CTMC is lumpable under *pop*-1 precisely because class 1 has preemptive priority over all other calls. The multi-class model where priorities are not preemptive is not lumpable under *pop*-1.

The *aop* is a valuable partition for the preemption model because the preemption probability can be expressed in terms of the probability of being in aggregate occupancy level c. Unfortunately, as we have seen, *aop* is only lumpable under homogenous service rates. The *pop*-1 is appealing as it is lumpable under heterogeneous service rates, but this is of less value than *aop* because the partition does not map easily to the performance metrics of interest, *i.e.*, the preemption probabilities and rates. Nonetheless, the *pop*-1 is still of value in computing performance, especially when a time-scale separation holds among the various classes.

A. Decomposability and time-scale separation

Whereas Lumpability refers to a partition where the transition across subsets is not state-dependent, decomposability refers to a partition where the transition rate across subsets is zero, *i.e.*, the chain is reducible. Thus decomposability is a special case of Lumpability. Both lumpable and decomposable may be relaxed to quasi-lumpable (QL) and nearly completely decomposable (NCD), respectively. A CTMC is said to be ϵ *quasi-lumpable* if $\mathbf{Q}$ can be decomposed as $\mathbf{Q} = \mathbf{Q}^- + \mathbf{Q}^\epsilon$ where $\mathbf{Q}^-$ is lumpable and the largest element in $\mathbf{Q}^\epsilon$ has absolute value no larger than ϵ. A CTMC is said to be *nearly completely decomposable* if the states may be arranged into blocks such that $\mathbf{Q} = \mathbf{Q}^+ + \mathbf{Q}^\delta$, where $\mathbf{Q}^+$ is block diagonal, and the norm of the off-diagonal transition rates, $\|\mathbf{Q}^\delta\|$ is the degree of coupling. The intuition for QL is that "most" transitions across subsets are state-independent, and the intuition for NCD is that "most" transitions are within (rather than across) subsets. Just as decomposability implies Lumpability, Dayar and Stewart have shown that NCD implies QL, but the inverse need not hold [21]. In other words, NCD is a stronger condition than QL. This is natural since QL asserts the transitions across the subsets have a simple form, whereas NCD asserts the transitions across the subsets may be effectively ignored.

The previous subsection identified the priority 1 occupancy partition as lumpable, but pointed out that this by itself is of limited value since the partition does not map easily to the computation of the performance metrics of interest, *i.e.*, the preemption rates and probabilities. We now establish that the priority 1 occupancy partition is NCD under a time-scale separation among classes. A thorough discussion of time-scale separation for *discrete* time Markov chains is given in the book by Yin and Zhang [22]; Reiman and Schmitt use time-scale separation for a multi-class *non-preemptive* load on a connection pool [23]. We now establish that the *pop*-1 is NCD under a time-scale separation among classes.

Definition 3: The arrival rates and service rates obey a high-slow low-fast (*hslf*) time-scale separation *if*

$$\lambda_1 \ll \cdots \ll \lambda_K, \;\; \mu_1 \ll \cdots \ll \mu_K. \tag{13}$$

They obey a high-fast low-slow (*hfls*) time-scale separation *if*

$$\lambda_1 \gg \cdots \gg \lambda_K, \;\; \mu_1 \gg \cdots \gg \mu_K. \tag{14}$$

VI. Relationship to the optimal admission control

In the previous sections, we present our proposal and analyze the proposal through a Markov chain model. In this section, we show why we propose this certain preemption policy. It is well known that admission control is optimized through threshold policy. An immediate question of our proposal is what is the relationship of our proposed preemption policy to the optimal admission control? The following proposition shows under our proposal, the optimal admission control is still threshold-type. In a word, our proposal is suitable to the existing admission control policy.

Proposition 2: Under our preemption policy – only preempts lower priority connections when the pool is full, the coordinate convex admission control policy space equals the threshold admission control policy space: $\Pi_{\text{cc}}^a = \Pi_{\text{th}}^a$.

Proof: It is simple to verify that a threshold policy is coordinate convex. It remains to show that a coordinate convex policy is a threshold policy, or, equivalently, a non-threshold policy is not coordinate convex. Let Ω be the set of achievable states of a coordinate convex policy – we will show that admissions under a non-threshold policy violate the rules for Ω. A non-threshold policy must have two distinct states in one of the following two scenarios. If no two such states exist then the policy is of threshold type. See Fig. 10.

- Consider states n, n' be with $n_2 < n_2'$, $n_1 + n_2 < c$, and $n_1' + n_2' < c$ such that $\pi^a(n) = 0$ and $\pi^a(n') =$

1. Suppose $n_1 \leq n_1'$ (Fig. 10 left). Observe that $i)$ $n' \in \Omega$ and $ii)$ $n + e_2 \notin \Omega$ (due to $\pi^a(n) = 0$). But such a set is not coordinate convex by repeated application of the requirement $n' \in \Omega$ with $n_k' > 0$ implies $n' - e_k \in \Omega$. Suppose instead $n_1 > n_1'$ (Fig. 10 middle). Observe that $i)$ $\pi^a(n) = 0$ so that $n+e_2 = (n_1, n_2+1) \notin \Omega$. However, by repeating application of the requirement $n' \in \Omega$ with $n_2' > 0$ implies $n' - e_2 \in \Omega$, we obtain $(n_1', n_2+1) \in \Omega$. Notice $n_1' < n_1,\ n_1' + n_2 + 1 < n_1 + n_2 + 1 \leq c$, we admit class 1 calls from state $(n_1', n_2 + 1)$ till occupancy sum equals c, which means $n + e_2 \in \Omega$, which is a contradiction.

- Consider states n, n' with $n_2 = n_2'$, $n_1 < n_1'$, and $n_1' + n_2' < c$ such that $\pi^a(n) + \pi^a(n') = 1$ (Fig. 10 right). If $\pi^a(n) = 0, \pi^a(n') = 1$, then $n' + e_2 \in \Omega$ and $n + e_2 \notin \Omega$. Repeating application of the requirement $n' + e_2 \in \Omega$ with $n_1' > 0$ implies $n' + e_2 - e_1 \in \Omega$, we obtain $n + e_2 \in \Omega$, which is a contradiction. If $\pi^a(n) = 1, \pi^a(n') = 0$, then $n + e_2 \in \Omega$ and $n' + e_2 \notin \Omega$. However, Assumption **??** implies $n' + e_2 \in \Omega$ due to $n + e_2 \in \Omega,\ n_1 < n_1'$, which is a contradiction.

■

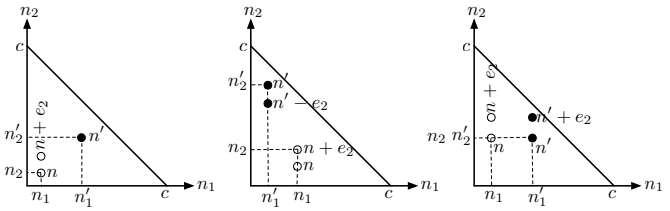

Fig. 10. Three cases of non-threshold policies discussed in the proof of Prop. 2.

We have the following relationships among the admission control policies and spaces:

$$\pi^a_{\rm cs} \in \Pi^a_{\rm th} = \Pi^a_{\rm cc} \subseteq \Pi^a_{\rm mf}. \tag{15}$$

Thus for any performance objective, say g, that depends upon the admission control policy π^a we have:

$$g_{\pi^a_{\rm cs}} \leq \max_{\pi \in \Pi^a_{\rm th}} g_\pi = \max_{\pi \in \Pi^a_{\rm cc}} g_\pi \leq \max_{\pi \in \Pi^a_{\rm mf}} g_\pi. \tag{16}$$

The restriction to coordinate convex policies (equivalently, here, threshold policies) may preclude achieving the overall optimal reward rate. For example, it would appear quite natural to consider a class of "sum rate threshold policies" (none of which is coordinate convex) where $\pi^a(n) = 1$ for $n_1 + n_2 < \tau \leq c$ for some $\tau \in [c]$. It is worth noting, however, that restriction to coordinate convex policies is common in the loss network admission control literature [24], [25], [26].

VII. Simulation results

Consider a single IL with $c = 100$, $K = 2$, arrival rates λ_1, λ_2 (to be varied), and $\mu_1 = \mu_2 = 1$ (homogeneous service rates). **Fig. 11** contains a plot of preemption probabilities versus r. The preemption probabilities are obtained from the rate expressions by dividing by the appropriate arrival rate: the preemption probability for class 1 is P_1/λ_1. The probability is to be understood as a "customer" average, *e.g.*, P_1/λ_1 is the fraction of arriving class 1 calls that cause a preemption. Further, each curve is actually a superposition of simulation results, exact numerical results (from §IV).

Fig. 11 presents P_1/λ_1 where $\lambda_1 = r$, and λ_2 is varied among $10r$, r, and $0.1r$. In each case the preemption probability is seen to be increasing, reach a maximum very near to $\lambda_1 = \rho_1 = c_1 = 100$, and then be convex decreasing. The initial increase is because increasing λ_1 moves the link from an underloaded regime to an overloaded regime: the number of preemptions increases as the system "fills up". The subsequent decrease is because as λ_1 continues to increase, it is increasingly likely that *all* circuits are occupied by class 1 calls, and thus arriving class 1 calls are blocked, rather than admitted by preempting a class 2 call. P_1/λ_1 is increasing as λ_2 increases from $0.1\lambda_1$ to λ_1 to $10\lambda_1$: a higher λ_2 means there are more class 1 arrivals that preempt class 2 calls.

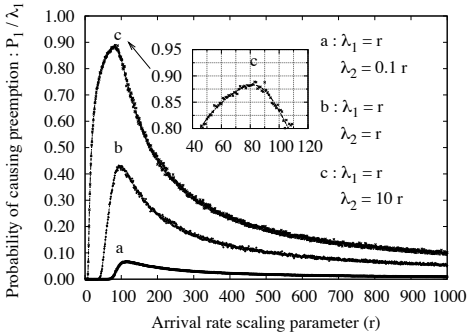

Fig. 11. Single link with $K = 2$ classes and homogeneous service rates. Preemption probabilities versus the arrival rate scaling parameter r. **Top:** P_1/λ_1 versus r; **Middle:** Q_2/λ_2 versus r; **Bottom:** P_1/λ_1 and Q_2/λ_2 versus r.

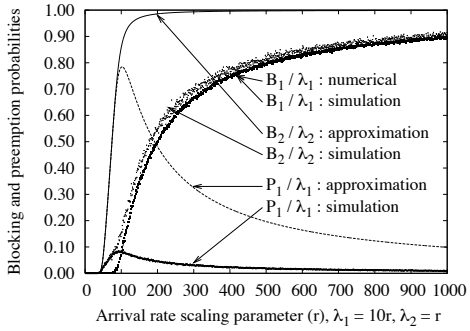

Fig. 12. Single link with $K = 2$ classes and homogeneous service rates. Preemption probabilities versus the arrival rate scaling parameter r. **Top:** P_1/λ_1 versus r; **Middle:** Q_2/λ_2 versus r; **Bottom:** P_1/λ_1 and Q_2/λ_2 versus r.

Consider a connection pool with $c = 100$ servicing $K = 2$ classes with two different settings for $\lambda_1, \lambda_2, \mu_1, \mu_2$:

scaling	λ_1	λ_2	μ_1	μ_2	ρ_1	ρ_2
hfls	$10r$	r	10	1	r	r
hslf	$0.1r$	$10r$	0.1	10	r	r

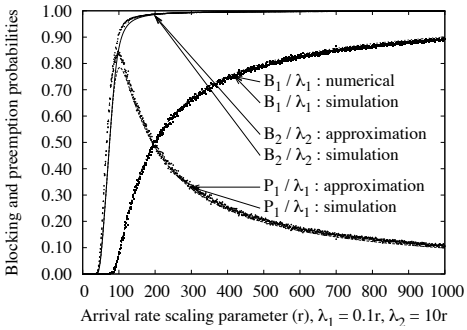

Fig. 13. Single link with $K = 2$ classes and homogeneous service rates. Preemption probabilities versus the arrival rate scaling parameter r. **Top:** P_1/λ_1 versus r; **Middle:** Q_2/λ_2 versus r; **Bottom:** P_1/λ_1 and Q_2/λ_2 versus r.

The first scaling corresponds to a *hfls* time-scale separation, and the second to a *hslf* time-scale separation. Note that the offered loads for the two classes are equal for both scalings, *i.e.*, $\rho_1 = \rho_2$. The fact that $\mu_1 \neq \mu_2$ means we have heterogeneous service rates. **Fig. 12** presents numerical and simulation results of the preemption probability P_1/λ_1 and the blocking probabilities B_1/λ_1 and B_2/λ_2 versus r. The **top** figure demonstrates the inaccuracy of the NCD approximations in the scaling of high priority fast, low priority slow, while the **bottom** figure shows the NCD approximation to be accurate in the scaling of high priority slow, high priority fast.

VIII. Conclusion

In this paper, we propose to apply preemption policy to the http connection pool so the web-based applications within our company can collaborate with each other when the resource is almost filled up. After modeling and analyzing the performance of this proposal, we derive the close-form of characterizations of our proposed preemption policy and show the model limitations to it. Moreover, we show our proposal fit the optimal admission control which allows the organization that has already employed the optimal admission control to their applications to use our preemption policy. Our numerical and simulation results show the model works fine with homogeneous service rate and also show under certain conditions the model works fine (not good) with heterogeneous service rates.

References

[1] Frank Kelly, "Loss Networks," *Annals of Applied Probability*, vol. 1, pp. 319–378, 1991.

[2] J.A. Garay and I.S. Gopal, "Call preemption in communication networks," in *Proceedings of IEEE INFOCOM*, Florence, Italy, May 1992, vol. 3, pp. 1043–1050.

[3] M. Peyravian and A.D. Kshemkalyani, "Connection preemption: issues, algorithms, and a simulation study," in *Proceedings of IEEE INFOCOM*, Kobe, Japan, April 1997, vol. 1, pp. 143–151.

[4] J.C. de Oliveira, C. Scoglio, I.F. Akyildiz, and G. Uhl, "New preemption policies for DiffServ-aware traffic engineering to minimize rerouting in MPLS networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 733–745, August 2004.

[5] J. Sung-eok, R.T. Abler, and A.E. Goulart, "The optimal connection preemption algorithm in a multi-class network," in *Proceedings of IEEE International Conference on Communications (ICC)*, New York, NY, April 2002, vol. 4, pp. 2294–2298.

[6] S. Tong, D. Hoang, and O. Yang, "Bandwidth allocation and preemption for supporting differentiated-service-aware traffic engineering in multi-service networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, New York, NY, April 2002, vol. 2, pp. 1305–1309.

[7] V. Stanisic and M. Devetsikiotis, "A dynamic study of providing quality of service using preemption policies with random selection," in *Proceedings of IEEE International Conference on Communications (ICC)*, Anchorage, AK, May 2003, vol. 3, pp. 1543–1546.

[8] F. Blanchy, L. Melon, and G. Leduc, "Routing in a MPLS network featuring preemption mechanisms," in *Proceedings of the International Conference on Telecommunications (ICT)*, Tahiti, Papeete, French Polynesia, February 2003, vol. 1, pp. 253–260.

[9] K. Yu, L. Zhang, and H. Zhang, "A preemption-aware path selection algorithm for DiffServ/MPLS networks," in *Proceedings of the IEEE Workshop on IP Operations and Management*, Beijing, China, October 2004, pp. 129–133.

[10] R.C. Vieira and P.R. Guardieiro, "A proposal and evaluation of a LSP preemption policy implemented with fuzzy logic and genetic algorithms in a DiffServ/MPLS test-bed," in *Proceedings of International Conference on Communications, Circuits and Systems*, Hong Kong, China, May 2005, vol. 1, pp. 109–114.

[11] Keith W. Ross, *Multiservice loss models for broadband communication networks*, Springer Verlag, 1995.

[12] R. Srikant, *The mathematics of Internet congestion control*, Birkhäuser, Boston, MA, 2003.

[13] H. White and L.S. Christie, "Queuing with preemptive priorities or with breakdown," *Operations Research*, vol. 6, no. 1, pp. 79–95, Jan.–Feb. 1958.

[14] D.R. Miller, "Computation of steady-state probabilities for $M/M/1$ priority queues," *Operations Research*, vol. 29, no. 5, pp. 945–959, Sep.–Oct. 1981.

[15] J.P. Buzen and A.B. Bondi, "The response times of priority classes under preemptive resume in $M/M/m$ queues," *Operations Research*, vol. 31, no. 3, pp. 456–465, May–June 1983.

[16] B. Ngo and H. Lee, "Analysis of a pre-emptive priority $M/M/c$ model with two types of customers and restriction," *Electronics Letters*, vol. 26, no. 15, pp. 1190–1192, July 1990.

[17] Y.Z. Cho and C.K. Un, "Analysis of the $M/G/1$ queue under a combined preemptive/nonpreemptive priority discipline," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 132–141, January 1993.

[18] W. Helly, "Two doctrines for the handling of two-priority traffic by a group of N servers," *Operations Research*, vol. 10, no. 2, pp. 268–269, Mar.–Apr. 1962.

[19] P.J. Burke, "Priority traffic with at most one queueing class," *Operations Research*, vol. 10, no. 4, pp. 567–569, Jul.–Aug. 1962.

[20] D.A. Calabrese, M.J. Fischer, B.E. Hoiem, and E.P. Kaiser, "Modeling a voice network with preemption," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 22–27, January 1980.

[21] T. Dayar and W.J. Stewart, "Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains," *SIAM Journal of Matrix Analysis and its Applications*, vol. 18, no. 2, pp. 482–498, April 1997.

[22] G.G. Yin and Q. Zhang, *Discrete-time Markov chains: two-time-scale methods and applications*, Springer, New York, NY, 2005.

[23] M.I. Reiman and J.A. Schmitt, "Performance models of multirate traffic in various network implementations," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunication Networks*, pp. 1217–1228. Elsevier, 1994.

[24] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Transactions on Communications*, vol. COM-29, no. 10, Oct. 1981.

[25] K. W. Ross and D. H. K. Tsang, "The stochastic knapsack problem," *IEEE Transactions on Communications*, vol. 37, no. 7, Jul. 1989.

[26] J. M. Aein, "A multi-user-class blocked-calls-cleared demand access model," *IEEE Trans. Commun.*, vol. COM-26, Mar. 1978.