



Call admission control for capacity-varying networks *

J. SIWKO ** and I. RUBIN

*56-125B Engineering IV, Electrical Engineering Department, University of California, Los Angeles,
Los Angeles, CA 90095-1594, USA*
E-mail: {siwko;rubin}@ee.ucla.edu

Abstract. Many networks, such as Non-Geostationary Orbit Satellite (NGOS) networks and networks providing multi-priority service using advance reservations, have capacities which vary over time for some or all types of calls carried on these networks. For connection-oriented networks, Call Admission Control (CAC) policies which only use current capacity information may lead to excessive and intolerable dropping of admitted calls whenever the network capacity decreases. Thus, novel CAC policies are required for these networks. Three such CAC policies are discussed, two for calls with exponentially distributed call holding times and one for calls whose holding time distributions have Increasing Failure Rate (IFR) functions. The Admission Limit Curve (ALC) is discussed and shown to be a constraint limiting the conditions under which any causal CAC policy may admit calls and still meet call dropping guarantees on an individual call basis. We demonstrate how these CAC policies and ALC represent progressive steps in developing optimal CAC policies for calls with exponentially distributed call holding times, and extend this process to the more general case of calls with IFR call holding times.

Key Words: call admission control (CAC), non-geostationary orbit satellite (NGOS), low earth orbit satellite (LEOS), capacity-varying networks, wireless networks, telecommunications

1. Introduction

Many networks have capacities which vary over time for many reasons. Some of these causes, such as a sudden partial failure, may not be predictable. Other causes, however, may be predictable. For example, consider a Non-Geostationary Orbit Satellite (NGOS) system where a given geographic area is served by a succession of beams and satellites. The capacity available to serve the area varies over time due to the power distribution of beams within a satellite, the reuse pattern of channels, the number of satellites and beams visible to the area, and whether the area is currently served by a central beam with a small footprint or an edge beam with a large footprint. These capacity variations over a given region in an NGOS system are cyclic, recurring as the same sequence of satellites complete their orbits to serve the area once again. Due to the different interactions listed above, the periodicity of the capacity variation sequence may be considerably longer than the periodicity of a satellite's orbit, but nonetheless the capacity variations are periodic and thus known in advance.

* This work was supported by SBC Pacific Bell and University of California MICRO Grants No. 96-157, 97-152, and 98-131, and ARO Grant No. DAAG55-98-1-0338.

** Now with Trilogy Software, Inc., 6034 W. Courtyard Dr., Austin, TX 78730, USA.

Predictable capacity variations occur in terrestrial networks as well. For example, consider a mobile base station which uses GPS data to determine its future terrain, and thus its future bandwidth capacity. Or, consider a multimedia network which accepts advance reservations for services such as a corporate videoconference or a pay-per-view broadcast. With these reservations guaranteed a high priority in advance, the network resources available to serve other, lower priority users will decrease at the start of the reserved time, and thus constitute a deterministic future capacity change from the perspective of these lower priority users. Many other examples also exist of networks whose capacities change in a deterministic manner with some amount of advance notice.

Due to scarcity of resources such as spectrum, Call Admission Control (CAC) policies are vital in a network's ability to guarantee Quality of Service (QOS) to connection-oriented services. CAC policies protect a network from overloading by determining whether incoming call requests should be accepted or rejected. As befits a subject of such importance, CAC policies have been the subject of considerable study (see [3–5,7,8] and the many references found in [6]).

Many of these proposed CAC policies can be described as making admission decisions by comparing the resources required by an incoming call request with the resources *currently* available in the network. These resources may include physical resources, such as bandwidth, or virtual resources, such as effective bandwidth. By only considering currently available resources, these policies implicitly assume that the network's capacity will remain constant over the time frame of any admitted call, that is, the network is *fixed-capacity*.

In capacity-varying networks, such as the examples listed earlier, a reduction in network capacity may affect calls in progress at that time. Under a CAC policy which considers only currently available resources, calls may be accepted prior to the known capacity change only to be dropped once the capacity decreases. A more intelligent CAC policy, aware of the upcoming capacity change, might block calls instead of accepting them and then dropping them. Dropping a call is generally considered a less desirable result than blocking a call request, since dropping a call involves breaching QOS guarantees made upon call acceptance, guarantees which were not made in blocking the call request.

Another topic of recent interest is the notion of "book-ahead" or advanced reservation of resources for calls. Several papers have proposed reservation and call admission policies [1,2,13], but they have focused on the admission of the reserving calls. Our work complements the previous work on reserving calls by proposing a CAC policy for the nonreserving calls which enables the network to allow reservations without undue dropping of nonreserving calls.

A CAC policy for a LEOS network was proposed in [12] which assumes the use of fixed direction satellite antennas. Consequently, the beam footprints move with respect to Earth in a continuous fashion. In contrast, when applied to NGOS networks, our work is applicable to satellites with dynamically steerable antennas. With these satellites, the antennas are constantly shifted in order to serve the same geographic region for longer

periods of time [14]. After the satellite moves out of position, the antennas are then swung to serve an entirely different geographic region. Thus, the beam footprints move with respect to Earth in a discrete fashion.

CAC policies for variable capacity networks for calls with exponentially distributed call holding times have appeared in [9–11]. This paper reviews these CAC policies, and ties them together as a sequence of progressive steps. This paper also presents a CAC policy representing the first step in this sequence for the more general case of calls whose call holding time distributions have Increasing Failure (or hazard) Rate (IFR) functions.

Given a call holding time distribution, the first step in this sequence is to find the optimal CAC policy for the Last Come First Dropped (LCFD) dropping policy. The LCFD dropping policy has the virtue of simplifying analysis, and is of practical interest as well. The next step is to use the structure of the optimal CAC policy for LCFD to find the conditions limiting admissibility under any conforming pair of CAC and dropping policies. From these limiting conditions one may be able to develop a nonconforming CAC/dropping policy pair which serves as a lower blocking bound among all conforming CAC policies, i.e., those which provide call dropping guarantees. The third step is to use the limiting conditions as a guide to develop conforming CAC/dropping policy pairs which admit calls under as many conditions as possible, and which thus minimize call blocking, while still meeting the call dropping guarantees.

Section 2 describes our model and gives definitions of dropping policies, CAC policies, and conforming pairs. Section 3 reviews the “CVCAC” policy, which is proven optimal for the Last Come First Dropped (LCFD) dropping policy. The concept of admissibility is defined in section 4. This section also reviews the Admission Limit Curve (ALC) and shows that it forms a tight constraint on the conditions under which any conforming CAC policy may accept a call request. As a result, using the ALC for admission decisions results in a lower bound on the blocking performance of any causal, conforming CAC policy.¹ Some of the ideas behind the ALC were used in the development of the Capacity-Varying Greedy Heuristic (CVGH) CAC policy appearing in section 5, which yields blocking performance close to this lower bound while still being conforming with respect to the Uniform Random Dropping policy. Simulation results comparing the performance of CVCAC, CVGH, and the naive “no CAC” policy are given in section 6. The CVCAC, ALC, and CVGH can be viewed as a sequence of progressive steps for calls with exponential call holding times. The “CVIFR” policy for calls with IFR call holding time distributions is described in section 7, along with a proof of its optimality for the LCFD dropping policy. The CVIFR can be viewed as an extension of the CVCAC policy, and thus as the first step of a sequence of optimal CAC policies for calls with IFR holding times. Finally, conclusions appear in section 8.

¹ Noncausal CAC policies can achieve better blocking performance, but they are not realistic for most network operations.

2. Model and policy definitions

This section defines several concepts and models which are used throughout this paper. An analytic model of a capacity-varying network is introduced in section 2.1. Precise definitions of dropping policies and CAC policies under this model are given in sections 2.2 and 2.3, respectively. Section 2.3 also defines the key concept of conformity.

2.1. Model description

Consider a multiple access network whose communication resources are shared among a multitude of stations. The network is connection-oriented, so stations desiring to use the network must first submit a call request to the network access controller. These call requests may be accepted or rejected. Each call request accepted by the controller results in the allocation of some network resources to service the newly-made connection or call. These resources may be physical resources, such as bandwidth, power, and buffer space, for a constant bit-rate circuit-switched call; or virtual resources such as effective bandwidth for a variable bit-rate call; or a combination of both.

Consider a class of homogeneous calls subject to variations in the amount of resources available to serve the class. The class may consist of a subset of calls, such as nonreserving calls in a network which accepts reservations, or may consist of all calls. To support n calls of this class simultaneously requires an amount $R(n)$ of resources, where $R(n)$ is a monotone, nondecreasing function. Also define the inverse function for a given amount of resources r as $R^{-1}(r) \equiv \max(n \mid R(n) \leq r)$.

Define the *capacity* of the system at a given time t as the maximum amount of resources it can allocate² at t , and the *system size* as the number of active calls. Call requests arrive at the system with mean arrival rate λ . A Call Admission Control (CAC) policy decides whether a call request is to be admitted or rejected. Calls whose requests are rejected by the CAC policy are said to be *blocked* and are lost. A CAC policy must block a call request if the call's admission would cause the allocated resources to exceed the capacity. Holding times of calls are independent random variables with mean holding times of $1/\mu$.

When viewed over time, the capacity forms a piecewise-constant function with discontinuities at Capacity Change Times (CCTs) $\{T_i\}$. The capacity is left-continuous, so any dropping associated with a capacity change time T_i would occur at T_i^+ , i.e., just after T_i . Capacity change times are assumed to occur sufficiently far apart that at any given time we need only consider the impact of the next capacity change. (CAC policies, like any other connection-level mechanism, are most effective on events occurring on the timescale of connection holding times. Capacity changes occurring more frequently may be more effectively controlled by lower layers, such as physical layer mechanisms.)

²This includes both resources which have been allocated to service active calls and unused resources which can be allocated to service new calls.

For a given time t , define the following variables:

$$\begin{aligned}
T &= \min(T_i \mid T_i \geq t) = \text{time of next capacity change,} \\
r_0 &= \text{system capacity at } t, \\
r_1 &= \text{system capacity at } T^+, \text{ i.e., just after } T, \\
C_0 &= R^{-1}(r_0) = \text{maximum number of active calls at } t, \\
C_1 &= R^{-1}(r_1) = \text{maximum number of active calls at } T^+, \\
\varepsilon &= \text{dropping probability threshold,} \\
N(t) &= \text{set of calls active at } t^-, \text{ i.e., just before } t, \\
n(t) &= \text{system size at } t^-, \text{ i.e., } |N(t)|, \\
\mathcal{H}(t) &= \text{history up to time } t.
\end{aligned}$$

To decide whether a call request arriving at a time t under a specific past history $\mathcal{H}(t)$ should be admitted or rejected, we generally need to consider the possible future events conditioned on the admission of this call request at t , and then determine whether this conditional future will result in dropping probabilities exceeding the threshold ε . Therefore, we define the following variables for a call request arrival time t :

$$\begin{aligned}
\widehat{\mathcal{H}}(t) &= \mathcal{H}(t) \cup \{\text{the call request at } t \text{ is accepted}\}, \\
\widehat{N}(t) &= N(t) \cup \{\text{the call arriving at } t\}.
\end{aligned}$$

If there is no call request arrival at t , then $\widehat{\mathcal{H}}(t)$ and $\widehat{N}(t)$ equal $\mathcal{H}(t)$ and $N(t)$, respectively. We also define the following symbols for operations:

$$\begin{aligned}
[c]^+ &= \max(0, c), \\
|X| &= \text{the cardinality of set } X, \text{ i.e., the number of elements in set } X.
\end{aligned}$$

The case of a capacity increase at T (i.e., $r_1 > r_0$) is straightforward, since any call admitted before T will continue to have adequate resources after T . The interesting case is a capacity decrease at T (i.e., $r_1 < r_0$). If the system size at T is larger than C_1 , then the system will no longer have the resources to support all the admitted calls and must drop some. For many applications, dropping a call after admission is considered much more disruptive and much less desirable than blocking the call, and thus it is imperative to carefully regulate call dropping. In particular, we assume the existence of a dropping probability threshold ε which forms part of the QOS guaranteed by the network to every admitted call. In other words, the network guarantees dropping probabilities on an individual call basis, just as it guarantees other common QOS parameters such as packet loss, delay, and jitter.

2.2. Dropping policies

The choice of which calls are dropped at a capacity change time is determined by a *dropping policy*. A dropping policy \mathcal{D} is any set of rules, possibly probabilistic and possibly depending on the past system history, which select $[|N(T)| - C_1]^+$ specific calls to drop out of the $N(T)$ calls active at T^- . Given a history $H = \mathcal{H}(T)$, which includes a set of calls $N = N(T)$, \mathcal{D} induces a dropping probability distribution on the space $\mathcal{V}(N)$ of all subsets of N of size $[|N| - C_1]^+$. Let $P_{\mathcal{D}}(V, N, H)$ be the probability that a subset $V \subset N$ is dropped under dropping policy \mathcal{D} and history H .

From this probability distribution, one can find for each call k the marginal probability³ $P_{\mathcal{D}}(k, N, H)$ that call k is one of the calls chosen to be dropped, by the summation

$$P_{\mathcal{D}}(k, N, H) = \sum_{V \ni k} P_{\mathcal{D}}(V, N, H). \quad (1)$$

These marginal probabilities have the property that they sum to a constant, namely,

$$\sum_{k \in N} P_{\mathcal{D}}(k, N, H) = \sum_{k \in N} \sum_{V \ni k} P_{\mathcal{D}}(V, N, H) = \sum_{V \in \mathcal{V}(N)} \sum_{k \in V} P_{\mathcal{D}}(V, N, H) \quad (2)$$

$$= \sum_{V \in \mathcal{V}(N)} |V| P_{\mathcal{D}}(V, N, H) = |V| = [|N| - C_1]^+ \quad (3)$$

since all subsets V have the same size.

For example, suppose $|N| - C_1 = y > 0$. The Uniform Random Dropping (URD) policy \mathcal{U} , which assigns each call in N an equal marginal dropping probability, results in set dropping probabilities of

$$P_{\mathcal{U}}(V, N, H) \equiv \frac{1}{\binom{C_1+y}{y}} \quad \forall V \in \mathcal{V}(N) \quad (4)$$

and marginal call dropping probabilities of

$$P_{\mathcal{U}}(k, N, H) \equiv \frac{y}{C_1 + y} \quad \forall k \in N. \quad (5)$$

The Last Come First Dropped (LCFD) policy \mathcal{L} , which drops calls in reverse order of acceptance, results in set dropping probabilities of

$$P_{\mathcal{L}}(V, N, H) \equiv \begin{cases} 1 & \text{if } V = \{k_1, k_2, \dots, k_y\}, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

and marginal call dropping probabilities

$$P_{\mathcal{L}}(k, N, H) \equiv \begin{cases} 1 & \text{if } k = k_1, k_2, \dots, k_y, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where k_i is the i th most recently admitted call in N .

2.3. Call admission control policies

A Call Admission Control (CAC) policy is a set of rules which determine whether any given call request is to be admitted or blocked. In general, this decision may depend on any past or current information in the history $\mathcal{H}(t)$, such as system size, call ages, interarrival times, etc. The decision may also depend on the parameters T and C_1

³ It should be clear from context whether a given dropping probability refers to a set of calls or an individual call, and thus there should be no confusion about the notation.

describing the future capacity change, the call departure rate parameter μ , and the parameter ε determining the dropping service guarantee. The primary goal of a CAC policy is to meet the specified dropping criteria. Once the dropping criteria have been met, the secondary goal of a CAC policy is to maximize throughput. In the case where all calls are statistically identical, maximizing throughput is equivalent to minimizing call blocking.

A CAC policy/dropping policy pair $(\mathcal{C}, \mathcal{D})$ is said to be *conforming* if the CAC policy \mathcal{C} never admits any call request when, given the dropping policy \mathcal{D} , the probability of dropping that call, or any currently active call, is greater than the dropping probability threshold ε . A conforming pair $(\mathcal{C}, \mathcal{D})$ thus guarantees dropping probabilities on an individual call basis, not just for the average of all calls. Given a dropping policy \mathcal{D} , a CAC policy \mathcal{C} is said to be *conforming with respect to \mathcal{D}* if $(\mathcal{C}, \mathcal{D})$ is conforming. Finally, a CAC policy \mathcal{C} is said to be *conforming* if there exists some dropping policy \mathcal{D} such that $(\mathcal{C}, \mathcal{D})$ is conforming.

To design a conforming CAC policy \mathcal{C} , one must be able to calculate the call dropping probability at any time t .⁴ The dropping probability for a particular call will depend on the dropping policy used, which in turn may depend on the entire system history. The dropping probability as calculated at t may depend on the composition of the calls active at T , which in turn depends on admissions after t by the CAC \mathcal{C} . Therefore, in its most general form, the dropping probability for a particular call k as calculated at a time t is

$$P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}, \mathcal{C}) \equiv \sum_H P(\mathcal{H}(T) = H, N(T) = N \mid \widehat{\mathcal{H}}(t), \mathcal{C}) P_{\mathcal{D}}(k, H, N). \quad (8)$$

To reduce ambiguity in referring to “dropping probabilities”, $P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}, \mathcal{C})$ will be designated as t -dropping probabilities, and $P_{\mathcal{D}}(k, H, N)$ will be designated as T -dropping probabilities. Note that, since at $t = T$ the t -dropping probability equals the T -dropping probability, there is no notational discrepancy. The t -dropping probability is the probability that a call will eventually be dropped at T , given that it is active at t , while the T -dropping probability is the probability that a call is dropped, given that it is active at T .

Since a conforming CAC policy must guarantee that the t -dropping probability threshold is met for all calls at any time that a call request is admitted, the conformity requirement can be stated as

$$\max_{k \in \widehat{N}(t)} P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}, \mathcal{C}) \leq \varepsilon \quad (9)$$

in order to be able to admit a call at t .

⁴ Note that this does not imply that the *implementation* of the policy will have to calculate dropping probabilities.

3. The capacity-varying call admission control (CVCAC) policy

This section describes the CVCAC, or Capacity Varying Call Admission Control policy for exponential holding times. The derivation of the CVCAC policy is presented in section 3.1, followed by a discussion of the impacts of various parameters on the policy. Finally, section 3.2 proves the optimality of CVCAC among CAC policies conforming with respect to \mathcal{L} .

3.1. CVCAC policy definition

The dropping policy assumed in development of the CVCAC policy is the LCFD policy \mathcal{L} of dropping calls in reverse order of acceptance. Under this dropping policy, a call admitted at a time t will be dropped only if any calls admitted after t have been dropped, and the system still needs to drop additional calls. This dropping policy minimizes the amount of wasted effort by protecting calls that have been “invested” with greater amounts of service. As a result of this dropping policy, if a call is admitted at a time t , then this call’s eventual dropping at T depends only on the behavior of calls in the system at t and is independent of the behavior of any call requests, whether admitted or not, arriving after time t .

Consider a call j admitted at a time $t < T$. Because call dropping is performed in reverse order of arrival, the dropping of this call depends only on these two factors:

- (1) whether the call admitted at t is itself still active at T , just before the capacity change, and
- (2) the departure process of calls admitted before t .

In particular, note that the t -dropping probability is independent of the CAC after t . Let p_t be the call survival probability of call j , i.e., the probability that a call active or admitted at t is still active at T . Clearly this event must occur in order for this call to be dropped. Since service times are exponentially distributed,

$$p_t = e^{-\mu(T-t)}. \quad (10)$$

In addition, exponential service times mean that the survival probability of calls active at t is also p_t . Therefore, given a system size of $n(t)$ calls just before the call request arrival at t , the number of these calls still active at T is binomially distributed with parameter p_t .

By conditioning on the two factors listed earlier, we have for any CAC policy \mathcal{C}

$$P_j(t, \widehat{\mathcal{H}}(t), \mathcal{L}, \mathcal{C}) = \begin{cases} p_t \sum_{x=C_1}^n \binom{n}{x} p_t^x (1-p_t)^{n-x} & \text{if } n \geq C_1, \\ 0 & \text{if } n < C_1. \end{cases} \quad (11)$$

Given a dropping probability threshold ε , we can find for every t the largest value of n such that $P_j(t, \widehat{\mathcal{H}}(t), \mathcal{L}, \mathcal{C}) \leq \varepsilon$. Define

$$M(t) \equiv \max\{n \mid P_j(t, \widehat{\mathcal{H}}(t), \mathcal{L}, \mathcal{C}) \leq \varepsilon\}. \quad (12)$$

The CVCAC policy is to accept a call request arrival at t iff $n(t) \leq \min\{C_0 - 1, M(t)\}$.

3.2. Optimality

By design, the CVCAC policy is conforming with respect to dropping in reverse order of acceptance, which preserves calls that have been “invested” with the most network effort. Thus, the CVCAC policy satisfies the dropping probability requirement. Once the dropping criterion has been met, the next goal of a CAC policy is to maximize the throughput. In the case where all calls are statistically identical, maximizing throughput is equivalent to minimizing the number of blocked calls. We prove that the CVCAC policy is optimal in the following sense:

Theorem 1. Suppose call holding times are exponentially distributed and CCT is deterministic. If the arrival process is Poisson, then the expected blocking under CVCAC is less than or equal to the expected blocking under any causal CAC policy conforming with respect to \mathcal{L} .

Proof. Suppose two systems, identical except for their CAC policies, are both fed by the same Poisson arrival process. Let system A use CVCAC, and system B use any other causal CAC policy conforming with respect to \mathcal{L} . Define $n_A(t)$ and $n_B(t)$ as their respective system sizes. Assume that both systems are initialized with identical calls (or no calls).

In the absence of any other constraint, the expected throughput of either system over any period of time $[a, b]$ is $\int_a^b \mu n(t) dt$. For Poisson arrivals, if a time period begins with both systems having the same number of i.i.d. calls, then the system with higher expected throughput over a time period will also have lower expected blocking during that time period.

Time can be divided into two types of periods: those with $n_A(t) > n_B(t)$ and those with $n_A(t) \leq n_B(t)$.⁵ Consider a period with $n_A(t) > n_B(t)$, noting that such periods begin with both system sizes equal. Furthermore, since both CAC policies are causal, they cannot make call admission decisions based on the exact call holding time, and thus the calls in both systems have residual holding times which are i.i.d. Since, in addition, no constraints are applicable during these time periods, the expected blocking over these periods for system A is less than or equal to the expected blocking for system B.

Now consider a period with $n_A(t) \leq n_B(t)$. Since call holding times are exponentially distributed and since the dropping policy is \mathcal{L} , equation (11) applies to both systems. Therefore, if system B admits a call at some time t , then, by the definition of $M(t)$, it must have $n_B(t) \leq M(t)$ in order to be conforming with respect to \mathcal{L} . But we then also have $n_A(t) \leq M(t)$, so system A would also admit the call. (This constraint is why the arguments in the previous paragraph do not apply to these time periods.) Since

⁵ Technically, those periods $[a, b]$ with the properties $\{n_A(a) = n_B(a), n_A(t) > n_B(t) \forall t \in (a, b), \text{ and either } n_A(b) = n_B(b) \text{ or } b = T\}$, and the complements of these periods.

system A admits every call admitted by system B during these time periods, the expected blocking for system A over these periods is also less than or equal to the expected blocking for system B.

Since the expected blocking for system A is less than or equal to the expected blocking for system B over every time period, the conclusion follows. \square

As noted earlier, CVCAC provides a dropping probability guarantee on an individual call basis. A CAC policy aiming for an *average* dropping probability threshold can always provide less blocking by keeping track of the number of calls likely to be dropped and, if it is near T and under its average dropping probability threshold, start admitting calls that will almost certainly be dropped. However, it is unlikely that users would appreciate such an “improvement”.

4. Admission limit curve (ALC)

This section presents the Admission Limit Curve (ALC) for exponentially distributed call holding times. The ALC defines the conditions under which every conforming CAC policy must reject an incoming call request. Consequently, the ALC can be used as the basis for a CAC policy which, although nonconforming, is useful nonetheless as a lower bound on the blocking performance achievable by any conforming CAC policy.

Section 4.1 defines the concept of admissibility and introduces the ALC. The proof that the ALC is the admissibility boundary appears in section 4.2. The nonconforming CAC \mathcal{A} is presented in section 4.3, along with the proof that it is a lower blocking bound on any conforming CAC policy. Section 4.3 also discusses some additional properties of the ALC.

4.1. Definition of ALC

Given the parameters C_1, T, μ , and ε , a point (t, n) with $-\infty < t < T$ is said to be *admissible* if there exists some conforming pair $(\mathcal{C}, \mathcal{D})$ such that a call request arriving at t with the system size n can be admitted by CAC policy \mathcal{C} . Points (t, n) with $-\infty < t < T$ which are not admissible, i.e., points such that there is no combination of dropping policy and CAC policy such that a call request can be admitted at t with n active calls without violating the dropping probability threshold ε for at least one call, are said to be *inadmissible*. The set of all inadmissible points is called the *inadmissible region*, and the set of all admissible points is called the *admissible region*.

To find the boundary of the inadmissible region, suppose we are given the parameters C_1, T, μ , and ε , and that we use the URD dropping policy \mathcal{U} . Suppose also that we are given a specific $t < T$ and consider a CAC policy \mathcal{C}_t which admits at s prior to t if $n(s) < n$ for some specific admission cutoff level n and admits no calls after t , i.e.,

during $s \leq t$ admit a call iff $n(s) \leq n$,
during $s > t$ admit no calls.

Given such a CAC policy, it is clear that blocking is minimized by increasing n . The maximum value of n is restricted by the requirement that the CAC policy be conforming with respect to \mathcal{U} . Under this CAC and dropping policy, the highest call dropping probability calculated at any possible call admission time is achieved by a call request arriving at $s = t$ with $n(t) = n$ calls already active. Because no further calls would be admitted after t , the t -dropping probability can be computed for each call k active at t as⁶

$$\begin{aligned}
P_k(t, \widehat{\mathcal{H}}(t), \mathcal{U}, \mathcal{C}_t) &= P(\text{call } k \text{ active at } T \mid \text{call } k \text{ active at } t) \\
&\times \sum_{x=C_1}^n P(x \text{ other calls will be active at } T \mid n \text{ other calls active at } t) \\
&\times P(\text{call } k \text{ is dropped} \mid \text{call } k \text{ active at } T, x \text{ other calls active at } T, \mathcal{U}) \\
&= p_t \sum_{x=C_1}^n \binom{n}{x} p_t^x (1-p_t)^{n-x} \frac{x+1-C_1}{x+1}, \tag{13}
\end{aligned}$$

where p_t is the call survival probability as defined in equation (10).

For each $t < T$, define $L(t)$ as the value of n which minimizes blocking (by maximizing n) while satisfying the conformity criterion (9) for the CAC policy \mathcal{C}_t with parameter t . Thus,

$$L(t) = \max_{\text{integer } n} \left(n \mid \sum_{x=C_1}^n \binom{n}{x} p_t^{x+1} (1-p_t)^{n-x} \frac{x+1-C_1}{x+1} \leq \varepsilon \right). \tag{14}$$

The locus of points $(t, L(t))$ is called the Admission Limit Curve, or ALC. It can readily be proven that equation (13) is monotonically increasing with n and t , and thus the ALC is monotonically non-increasing with t . This monotonicity can be used to show that the CAC policies \mathcal{C}_t are in fact conforming with respect to \mathcal{U} . Also note that, since n is discrete, the ALC is a left-continuous step function.

4.2. ALC as a limit boundary

The importance of the ALC lies in the following theorem:

Theorem 2. Suppose holding times are exponentially distributed and CCT is deterministic. Then, no conforming CAC policy may admit a call request arriving at a time t if the system size $n(t)$ is such that $n(t) > L(t)$.

To prove this theorem, suppose that a conforming CAC/dropping policy pair $(\mathcal{C}, \mathcal{D})$ is given and that a call request arrives at a time t with the system size $n(t) >$

⁶ Note how the CAC policy \mathcal{C}_t here is analogous to the dropping policy \mathcal{L} in section 3 in the sense of making the t -dropping probability independent of the behavior of calls arriving after t .

$L(t)$. By the conformity requirement (9), the call request can be admitted only if the maximum t -dropping probability for calls in $\widehat{N}(t)$ is less than or equal to ε , i.e., $\max_{k \in \widehat{N}(t)} P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}, \mathcal{C}) \leq \varepsilon$. Let \mathcal{D}_G be a modification of \mathcal{D} such that all calls admitted after t are dropped before any call in $\widehat{N}(t)$ is dropped. It is straightforward that calls in $\widehat{N}(t)$ cannot have a worse t -dropping probability under this new dropping policy.

Now let \mathcal{U}_G be a dropping policy which will drop all calls admitted after t before dropping any members of $\widehat{N}(t)$. If any members of $\widehat{N}(t)$ must still be dropped, then they will be chosen with equal probabilities among all members of $\widehat{N}(t)$ that are still active at T , i.e., by URD among the members of $\widehat{N}(t)$. It can be shown that, for members of $\widehat{N}(t)$ and any \mathcal{D} ,

$$\max_{k \in \widehat{N}(t)} P_k(t, \widehat{\mathcal{H}}(t), \mathcal{U}_G, \mathcal{C}) \leq \max_{k \in \widehat{N}(t)} P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}_G, \mathcal{C}). \quad (15)$$

Therefore, by equations (13) and (14) and by monotonicity, if $n(t) > L(t)$, then

$$\max_{k \in \widehat{N}(t)} P_k(t, \widehat{\mathcal{H}}(t), \mathcal{D}, \mathcal{C}) > \varepsilon. \quad (16)$$

Since $(\mathcal{C}, \mathcal{D})$ is a conforming pair, it must reject the call request. In other words, all points (t, n) with $n > L(t)$ are inadmissible. The interested reader is referred to [11] for more details.

By construction, given any single point $(t, L(t))$ of the ALC, there exists a conforming CAC policy, namely, the CAC policy \mathcal{C}_t described above, which can admit a call request at t with the system size $n(t) = L(t)$. Since each point of the ALC is thus admissible, and since equation (13) is monotonic, we obtain the following:

Corollary 1. The inadmissible region has a boundary, and that boundary is the ALC.

Thus, the ALC acts as a fundamental boundary on all conforming CAC policies under all dropping policies.

4.3. Additional properties of the ALC

Figure 1 illustrates a few examples of the ALC for different values of ε . All the curves assume $T = 0$, $\mu = 0.1$, and $C_1 = 20$. Viewing time backwards from T , the ALC approaches infinity at a time $T - \tau$ which is a finite distance earlier than T . It can be readily shown that $T - \tau = \max_{t < T} (p_t < \varepsilon)$, which yields the formula

$$\tau = \frac{\ln(\varepsilon)}{\mu}. \quad (17)$$

τ can be considered an upper bound on the prior notice which must be given of a future capacity decrease in order to be able to meet the call dropping requirement. The bound τ is typically a few average call holding times long. However, advance notice of only a single average call holding time is generally sufficient for most practical capacity levels and changes.

Active Calls

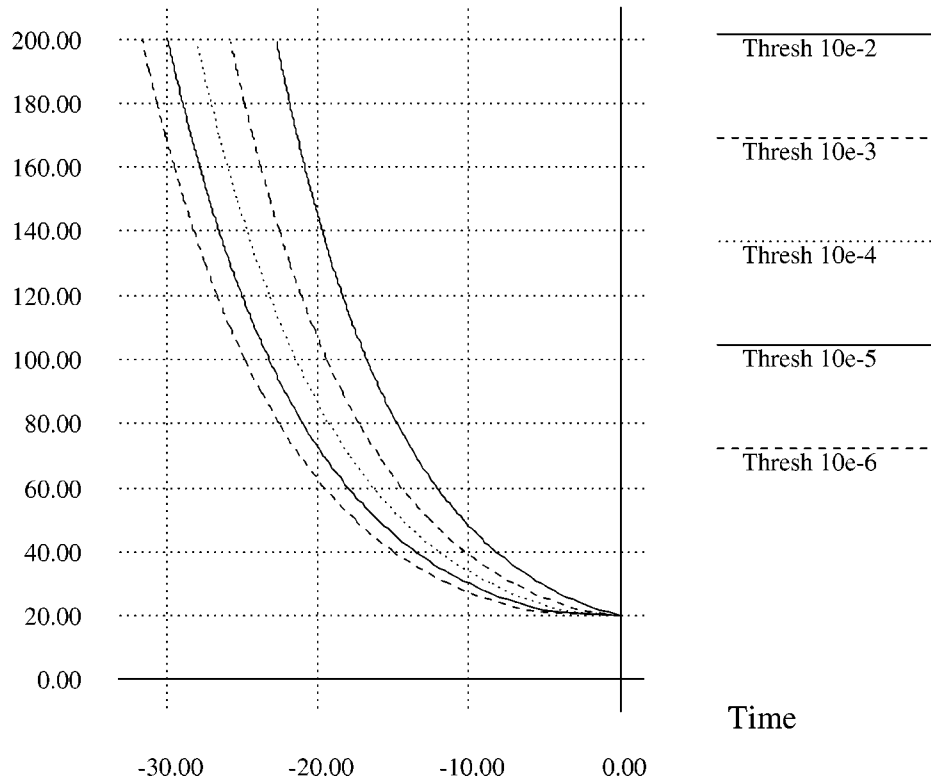


Figure 1. ALC boundaries.

The ALC could be used as the basis of a CAC policy \mathcal{A} , which admits a call request at a time t iff these two conditions are met:

- (1) $n(t) < C_0$,
- (2) $n(t) \leq L(t)$.

The appeal of policy \mathcal{A} lies in the following theorem.

Theorem 3. Suppose call holding times are exponentially distributed and the CCT is deterministic. If the arrival process is Poisson, then the expected blocking under CAC policy \mathcal{A} is less than or equal to the expected blocking under any conforming CAC policy.

Proof. As in the proof of theorem 1, suppose there are two systems, identical except for their CAC and dropping policies, both fed by the same Poisson arrival process. Let system A implement \mathcal{A} , and system B implement some conforming CAC policy \mathcal{C} .

During time periods when $n_A(t) > n_B(t)$, the expected blocking is lower in system A by the same arguments as in theorem 1.

Suppose now that $n_A(t) \leq n_B(t)$. If system B admits a call at some time t , then by theorem 2 we have $n_B(t) \leq L(t)$. Therefore, policy \mathcal{A} in system A will also admit this call. The rest of the proof follows as in the proof of theorem 1. \square

Unfortunately, policy \mathcal{A} is not conforming, as has been further confirmed through simulation. Thus, while it results in a lower bound on call blocking, it cannot be used to guarantee dropping performance.

5. The capacity-varying greedy heuristic (CVGH) CAC policy

Since the CAC policy \mathcal{A} based on the ALC is not conforming, this section discusses a CAC policy which does meet the dropping requirement and which achieves blocking performance close to that attained by policy \mathcal{A} . As in the previous section, calls have exponentially distributed holding times.

To minimize blocking, a conforming policy should seek to admit call requests in as much of the admissible region as possible. Since the admissible and inadmissible regions are defined in (t, n) space (given the parameters C_1, T, μ , and ε) it makes sense to construct a policy which bases admission decisions solely in (t, n) space as well (again, given the same 4 parameters.) Therefore, we construct a curve $K(t)$ in (t, n) space below the ALC, i.e., $K(t) \leq L(t) \forall t < T$. The Capacity-Varying Greedy Heuristic (CVGH, or \mathcal{V}) CAC policy is then defined to admit a call request at a time t iff both these conditions are met:

- (1) $n(t) < C_0$,
- (2) $n(t) \leq K(t)$.

The curve $K(t)$, like the ALC, is a left-continuous, decreasing, step function. Thus, $K(t)$ can be constructed from its discontinuous points. Let b_i be the unique time such that $K(b_i) = C_1 + i$ and $K(b_i^+) = C_1 + i - 1$. The discontinuous points b_i are found iteratively on the number of active calls $C_1 + i$. During the i th iteration, the greedy heuristic is used to maximize the value of b_i given the values of b_m for $m < i$ calculated in earlier iterations. The URD dropping policy is used, both because it is a commonly used dropping policy and because of the key role of URD in the proof of theorem 2. In addition, the call request arrival rate λ is assumed to be arbitrarily large ($\lambda \rightarrow \infty$) in order to construct a CAC policy which is conforming for any arrival rate.

The general iterative procedure is as follows: For a given i , a call request is assumed to arrive at b_i with the system size $n(b_i) = C_1 + i$, so the call would be admitted under CVGH. The system size probability density is then calculated at each of the previously computed times b_m for $m < i$, noting that at each time b_m the system size $C_1 + m + 1$ can be viewed as an ‘‘absorbing state’’ due to the infinite arrival rate assumption. Finally, a probability density for the system size at T is obtained, which is

Active Calls

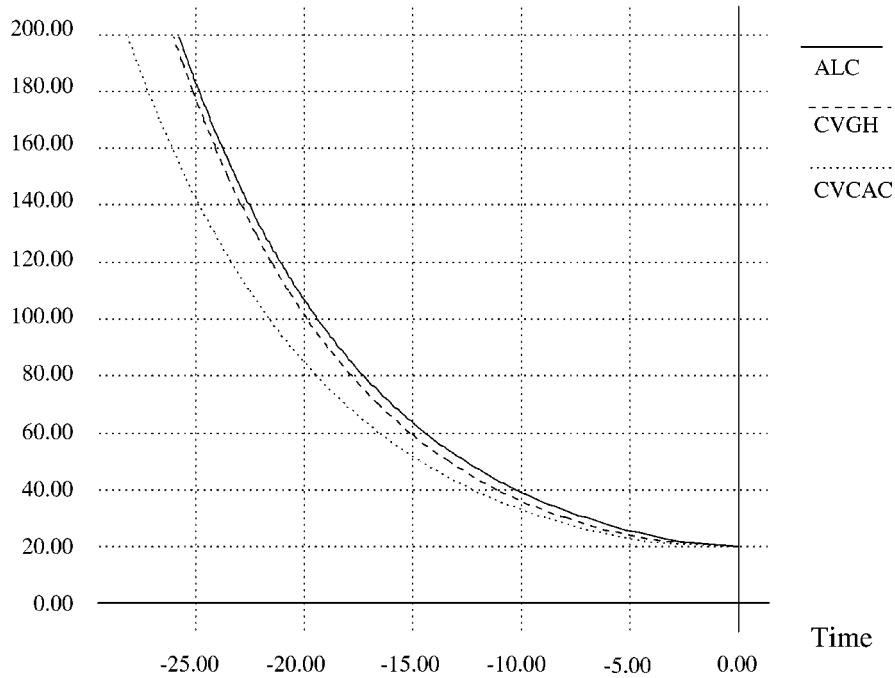


Figure 2. CAC boundaries.

then weighted by the appropriate $p_{\mathcal{U}}(k, N, \mathcal{H}(T))$ values to obtain an expression for the dropping probability calculated at b_i . This expression is then set equal to ε and solved by numerical methods to obtain the desired value of b_i . More details on this procedure may be found in [10,11].

Once these b_i values have been obtained, they may be stored in a lookup table for the implementation of the CVGH policy. The curve $K(t)$ is obtained from the b_i 's by $K(t) = C_1 + i \forall t \in (b_{i+1}, b_i]$ and $K(t) = C_1 - 1 \forall t \in (b_0, T]$.

Figure 2 displays the ALC and $K(t)$ curves along with the curve $M(t)$ used as the basis of the CVCAC policy. These curves are labeled ALC, CVGH, and CVCAC respectively. Active calls n is the dependent variable and time is the independent variable with $T = 0$. As can be seen, the CVGH curve lies very close to the limit curve ALC, and substantially closer to this limit than the CVCAC curve lies. Thus, we expect the CVGH policy to result in blocking performance close to the blocking lower bound of conforming CAC policies.

6. Simulation results for exponentially distributed call holding times

Simulations have been performed of systems implementing the CVCAC, CVGH and \mathcal{A} CAC policies. As a comparison, simulations were also performed using the “no

Dropping Probability

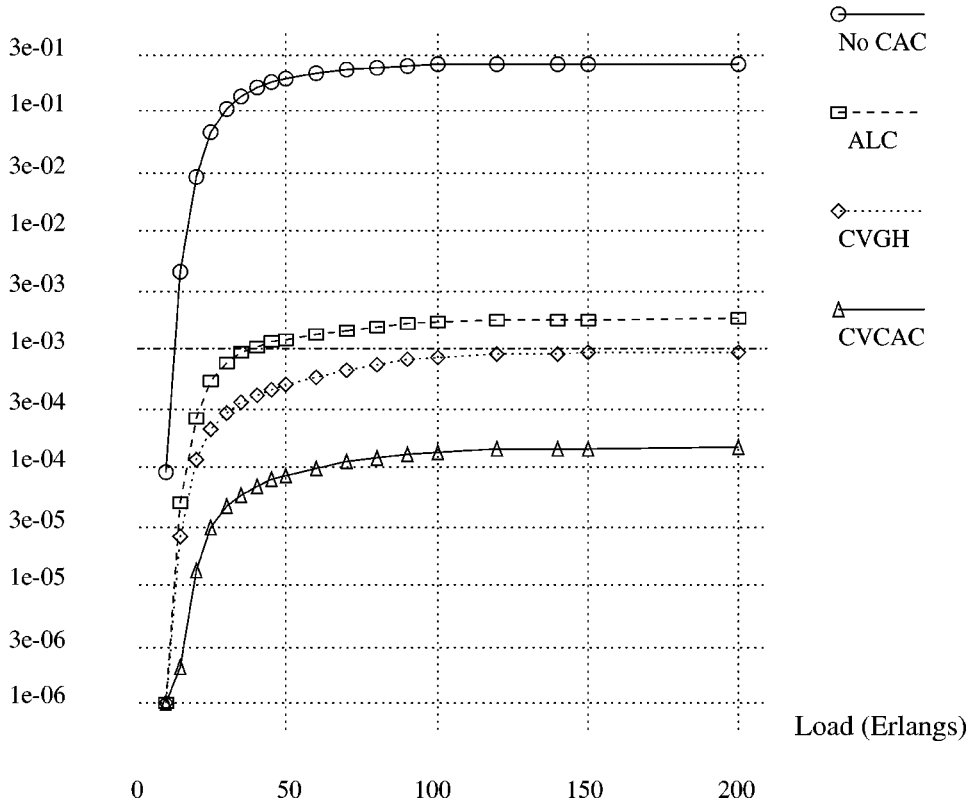


Figure 3. Dropping probabilities.

CAC” policy of always admitting as many calls as current capacity permits (i.e., admit if $n(t) < C_0$). All simulations began with a random number of active calls determined by the Erlangian or “steady-state” distribution of a system with a maximum capacity of C_0 calls. The time spanned was always 25 time units and contained one capacity change, a decrease, at the end of the simulation (i.e., $T = 25$). Dropping probabilities were obtained by dividing the number of calls dropped by the number of calls admitted during $[0, T]$ plus the number of initial active calls at time 0. Blocking probabilities were obtained by dividing the number of calls blocked by the number of call requests arriving during $[0, T]$.

Figure 3 displays dropping results and figure 4 displays blocking results for one sequence of simulations. For both figures, average call holding time is $1/\mu = 10$ time units per call, dropping probability threshold is $\varepsilon = 0.001$, current capacity serves a maximum of $C_0 = 100$ simultaneous calls, and the future capacity serves a maximum of $C_1 = 20$ simultaneous calls. The call request arrival rate λ , expressed as an Erlang load, is the independent variable.

Blocking Probability

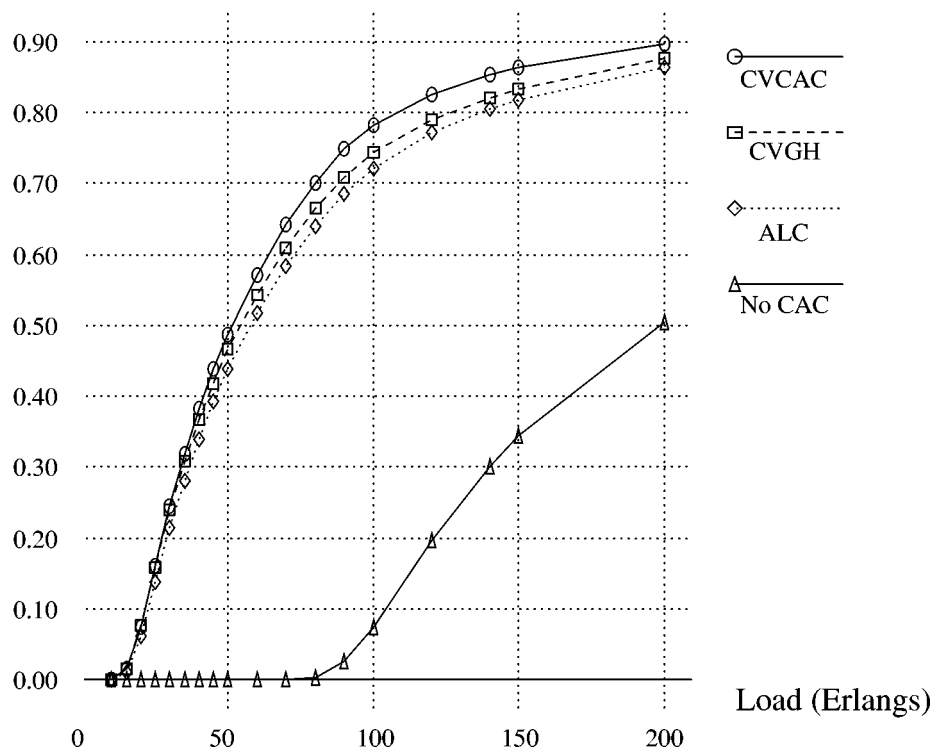


Figure 4. Blocking probabilities.

Dropping results are shown logarithmically in figure 3. As can be seen, both CVGH and CVCAC meet the dropping probability requirement, while \mathcal{A} does not, illustrating its non-conformity. CVGH is more efficient than CVCAC in terms of more closely approaching the dropping threshold level, suggesting a better tradeoff of less blocking for permitted dropping levels. The “no CAC” policy of admitting up to current capacity also does not meet the dropping criterion, and at higher loads can be orders of magnitude worse.

Blocking results are graphed in figure 4. At low offered loads, blocking probabilities are negligible for all policies. Blocking begins to rise rapidly at offered loads near the current call capacity C_0 for the “no CAC” policy and near the future call capacity C_1 for CVGH, CVCAC, and \mathcal{A} . Also, compared to CVCAC, CVGH results in blocking performance much closer to the lower blocking bound supplied by the nonconforming policy \mathcal{A} while still remaining conforming. Although blocking under CVGH, and under any conforming CAC policy for that matter, is higher than blocking without a CAC, this increase in blocking is a small price to pay for the greatly improved dropping performance.

7. The capacity-varying, increasing failure rate (CVIFR) CAC policy

This section describes the CVIFR, or Capacity-Varying, Increasing Failure Rate, CAC policy for Increasing Failure Rate (IFR) holding time distributions. Section 7.1 reviews the definition of an IFR distribution. The CVIFR policy is developed in section 7.2. CVIFR is a CAC policy guaranteeing call dropping probabilities on an individual call basis for dropping in reverse order of acceptance for any IFR holding time distribution. Simulation results comparing CVIFR to a CAC scheme which makes decisions only using current capacity information appear in section 7.3. Finally, section 7.4 discusses and proves the optimality of CVIFR among all CAC policies for IFR distributions providing call dropping probability guarantees on an individual call basis under the LCFD dropping policy.

7.1. Definition of increasing failure rate (IFR) distribution

Let

$b(t)$ = call holding time probability density,

$B(t)$ = call holding time cumulative distribution function.

Then the failure rate function (also known as the hazard rate function) is defined as

$$h(x) = \frac{b(x)}{1 - B(x)}.$$

The failure rate function $h(x)$ represents the conditional probability density that a call will end given that it has been in service for x time units. A holding time distribution is said to be an Increasing Failure Rate (IFR) distribution if $h(x)$ is a nondecreasing function of x . Examples of IFR distributions are uniform, exponential, gamma- n with $n \geq 1$, and half-Gaussian distributions.⁷

7.2. The CVIFR policy

Consider a call admitted at a time $t < T$, and suppose the LCFD dropping policy is used. Because call dropping is performed in reverse order of arrival, the dropping of this call depends only on these two factors:

- (1) whether the call admitted at t is itself still active at T , just before the capacity change, and
- (2) the departure process of calls admitted before t .

Suppose that $n(t)$ calls were active just prior to a call request arrival at t . Let

- s_k be the age or backwards recurrence time of the k th of these calls,
- \vec{s} be the $n(t)$ dimensional vector of these ages, and

⁷ The half-Gaussian random variable is the absolute value of a zero-mean Gaussian random variable.

- p_k be the survival probability, i.e., the probability that the k th call will still be active at T .

By using Bayes' theorem, we obtain

$$p_k = \frac{1 - B(T - t + s_k)}{1 - B(s_k)}. \quad (18)$$

This formula would also apply to the call request arriving at t , which has $s_{n(t)+1} = 0$, if the request is admitted. Therefore,

$$p_{n(t)+1} = 1 - B(T - t). \quad (19)$$

Since the call holding time distribution is IFR, if $s_i < s_j$ then $p_i \geq p_j$. Therefore, since the dropping policy is LCFD, $\max_{k \in \widehat{N}(t)} P_k(t, \widehat{H}(t), \mathcal{L}, \mathcal{C})$ occurs for the call with the lowest age, in particular, for the newly arriving call request at t . To simplify notation in this section, let $P_D(t | \vec{s})$ be the t -dropping probability of a call admitted at time t given that the ages of active calls at t^- are \vec{s} . By conditioning on the two factors listed above, we obtain

$$P_D(t | \vec{s}) = (1 - B(T - t)) \sum_{i_1=l_1}^{u_1} \cdots \sum_{i_j=l_j}^{u_j} \cdots \sum_{i_{n(t^-)}=l_{n(t^-)}}^{u_{n(t^-)}} \prod_{k=1}^{n(t^-)} [p_k^{i_k} (1 - p_k)^{u_k - i_k}], \quad (20)$$

where

$$u_j = \begin{cases} 0 & \text{if } C_1 - i_1 - i_2 - \cdots - i_{j-1} = 0, \\ 1 & \text{otherwise,} \end{cases} \quad (21)$$

and

$$l_j = \begin{cases} 0 & \text{if } C_1 - i_1 - \cdots - i_{j-1} - n(t^-) + j \leq 0, \\ 1 & \text{otherwise.} \end{cases} \quad (22)$$

A close examination of the definitions of u_j and l_j reveals that the expression $p_k^{i_k} (1 - p_k)^{u_k - i_k}$ can only take the values p_k or $1 - p_k$, which correspond to call k being active or not active at T . Thus, $P_D(t | \vec{s})$ is essentially obtained by summing over all combinations, among all calls active at t^- , of possible futures at T which lead to dropping of the call request.

Given a dropping probability threshold ε , the CVIFR policy is to admit a call request arriving at t iff $P_D(t | \vec{s}) \leq \varepsilon$ and $n(t) < C_0$.

7.3. Simulation results

Simulations have been performed of systems implementing the CVIFR policy and, for comparison, of systems employing the "naive" CAC policy of always admitting calls up to current available capacity. All simulations began with a random number of active

Dropping Probability

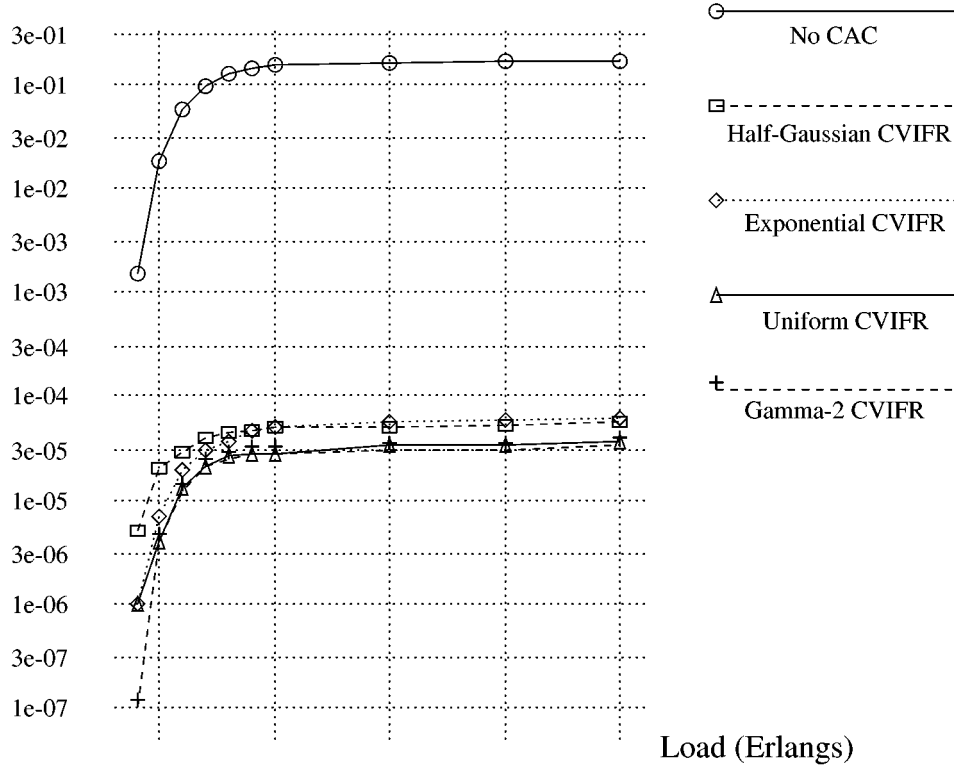


Figure 5. Dropping probabilities.

calls determined by the stationary distribution of an $M/G/C_0/C_0$ queue. The age of each initial call was randomly determined using the backwards recurrence time density

$$P\{\text{age} = s\} = \mu(1 - B(s)). \quad (23)$$

The residual holding time of each initial call was randomly determined using the density

$$P\{\text{residual holding time} = x \mid \text{age} = s\} = \frac{b(x + s)}{1 - B(s)}. \quad (24)$$

The time spanned was always 20 time units and contained one capacity change, a decrease, at the end of the simulation (i.e., $T = 20$ in all simulations.) Results comparing dropping and blocking probabilities for systems with and without the CVIFR policy are graphed in figures 5 and 6. Each data point in these graphs is the average of 100,000 simulations. Dropping probabilities were obtained by dividing the number of calls dropped by the number of calls admitted during $[0, T]$ plus the number of initial

Blocking Probability

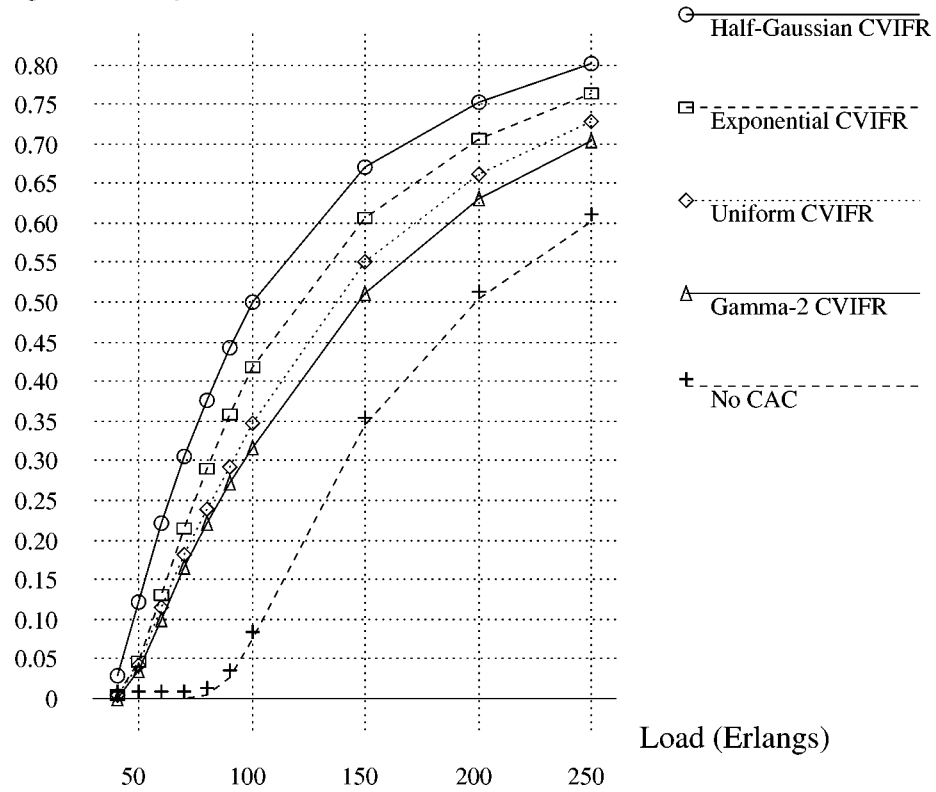


Figure 6. Blocking probabilities.

active calls at time 0. Blocking probabilities were obtained by dividing the number of calls blocked by the number of call requests arriving during $[0, T]$.

Both figures use the following parameters: dropping probability threshold $\varepsilon = 0.001$, current capacity $C_0 = 100$ calls, and future capacity $C_1 = 50$ calls. The call request arrival rate λ , expressed as an Erlang load, is the independent variable. The following holding time densities were used:

$$\begin{aligned}
 \text{uniform} & \quad b(x) = \mu/2, & 0 \leq x \leq 2/\mu, \\
 \text{exponential} & \quad b(x) = \mu e^{-\mu x}, & x \geq 0, \\
 \text{gamma-2} & \quad b(x) = 4\mu^2 x e^{-2\mu x}, & x \geq 0, \\
 \text{half-Gaussian} & \quad b(x) = \frac{2}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}, & x \geq 0, \\
 & \quad \text{with } \sigma = \frac{1}{0.4769\sqrt{2}\mu}.
 \end{aligned}$$

Each of these densities had an average holding time of $1/\mu = 10$ time units/call.

As can be seen in figure 5, the dropping probability threshold ε is consistently met by the CVIFR policy, even at quite high offered loads. In general, the dropping under the CVIFR policy increases slightly as the variance of the holding time distribution increases, but the difference is negligible. In contrast, the comparison system only meets the dropping probability threshold at low offered loads. At higher loads the dropping probability for the comparison system rises rapidly, finally leveling off at a value dependent on the current and future capacities, and on the length of the measuring period.

As illustrated by figure 6, blocking probability is low for both policies at low offered loads. Blocking begins to rise rapidly at offered loads near the future capacity C_1 for the CVIFR policy and near the current capacity C_0 for the comparison policy. In general, blocking under the CVIFR policy seems to increase as the variance of the holding time distribution increases. For all the holding time distributions simulated, the CVIFR policy has somewhat higher blocking probabilities than the comparison policy. However, as noted earlier, this increase in blocking probability is a small price to pay for the greatly improved dropping performance of the CVIFR policy.

7.4. Optimality

By design, the CVIFR policy is conforming with respect to dropping in reverse order of acceptance, which preserves calls that have been “invested” with the most network effort. Thus, the CVIFR policy satisfies the dropping probability requirement. Once the dropping criterion has been met, the next goal of a CAC policy is to maximize the throughput. In the case where all calls are statistically identical, maximizing throughput is equivalent to minimizing the number of blocked calls.

Define a comparison supersystem consisting of two systems, system 0 and system 1, side-by-side. Both systems are fed by a common Poisson arrival process $A(t)$, and both systems are identical except for their CAC policies. An infinite quantity of holding times are pre-generated using the holding time distribution $B(t)$ and these holding times are stored in a list. Each system has an identical copy of this list. Whenever a call request is admitted by a system, the holding time of the call in that system is the next number in that system’s copy of the list. Thus the i th call admitted to system 0 will have exactly the same holding time as the i th call admitted to system 1 *regardless of whether or not they correspond to the same call request arrival*.

For example, suppose that both systems have admitted $k - 1$ calls by t_1 , and the l th call request arrives at t_1 . Suppose further that this call request is accepted by system 1 and rejected by system 0. Suppose then that the $(l + 1)$ th call request arrives at t_2 and is accepted by both systems. Then the holding time of the call corresponding to the $(l + 1)$ th call request would be the k th list element in system 0 and the $(k + 1)$ th list element in system 1. Note also that the holding time in system 0 of the call accepted at t_2 would equal the holding time of the call in system 1 accepted at t_1 , not the holding time of the call in system 1 accepted at t_2 .

Both systems begin at time $t = 0$ with no calls (or w.l.o.g. with identical calls). Each system implements a different *causal* CAC policy, where a CAC policy is considered *causal* if it makes admission decisions using only these criteria:

- (1) past history,
- (2) call arrival and departure distributions,
- (3) information about the future capacity change, e.g., T , C_1 , and ε .

Note that although the systems possess complete knowledge of all future call holding times, this information is not available to the CAC policy. Also note that the CVIFR policy is causal since it makes admission decisions based on the current number of active calls and their ages, which encapsulates the past history; on the call holding time distribution; and on the future capacity change information.

Define

$$\begin{aligned} t_0^k &= \text{arrival time of } k\text{th call admitted to system 0,} \\ t_1^k &= \text{arrival time of } k\text{th call admitted to system 1,} \\ n_0(t) &= \text{number of active calls (system size) in system 0 at time } t, \\ n_1(t) &= \text{number of active calls (system size) in system 1 at time } t, \\ \vec{s}_0(t) &= \text{vector of ages of active calls in system 0 at time } t, \\ \vec{s}_1(t) &= \text{vector of ages of active calls in system 1 at time } t. \end{aligned}$$

Systems 0 and 1 are said to be *semi-equal* at time t if both systems have admitted the same number of calls during $[0, t)$. Systems 0 and 1 are said to be *equal* at time t if they are semi-equal, have the same system sizes, and each call in one system has the same age as a call in the other system.

Theorem 4. The number of calls blocked by CVIFR in this comparison supersystem is less than or equal to the number of calls blocked by any causal CAC policy conforming with respect to LCFD.

Proof. Without loss of generality, place the CVIFR policy in system 0 of the comparison supersystem, and place any other causal CAC policy \mathcal{C} conforming with respect to LCFD in system 1. The systems begin in equality. If both policies result in the exact same admission decisions for every call request, i.e., $t_0^i = t_1^i \forall i$, then both systems block the same number of calls and the theorem holds.

Suppose then that the policies result in different admission decisions. Let $i = \min_k \{t_0^k \neq t_1^k\}$ be the first call admitted at different times by the two systems, and let $t_\alpha = \min(t_0^i, t_1^i)$. If the call request arriving at t_α is rejected by the CVIFR policy in system 0 then, by construction of the CVIFR policy, the t -dropping probability of this call, should it be admitted, is greater than the dropping probability threshold ε . Since the systems are equal at t_α^- , since the call holding time distributions are IFR, and since Policy \mathcal{C} is conforming with respect to \mathcal{L} , system 1 could not admit such a call request. Therefore, the first time that the two policies act differently on a call request must be an admission by system 0 and a rejection by system 1. Thus, $t_\alpha = t_0^i$ and $t_1^i > t_0^i$.

In order for system 1 to admit more calls than system 0, semi-equality must first be achieved. If semi-equality is not achieved, then clearly system 0 admits more calls than system 1 for all times t throughout $[t_\alpha, T)$ and so the theorem holds. Let

$j = \min\{k \geq i \mid t_1^k < t_0^{k+1}\}$ be the call whose admission by system 1 results in semi-equality,

$t_s = t_1^j$ be the first time semi-equality is achieved after t_α .

We note that $\forall k \in [i, j]$ we must have $t_0^k < t_1^k$ since otherwise semi-equality would have been achieved for a call earlier than call j , contradicting the definition of call j . Therefore, for each admitted call k active in system 0 at t_s^+ (just after t_s), we have the corresponding admitted call k still active in system 1, so $n_0(t_s^+) \leq n_1(t_s^+)$. Furthermore, the age of each call k in System 0 is greater than or equal to the age of call k in system 1. Thus, if the age vectors are arranged so that calls active in both systems are first, in order of admittance times, followed by calls only active in system 1 (if any), then $\vec{s}_0(t_s^+) \geq \vec{s}_1(t_s^+)$. Moreover, this inequality remains true for all time t at least until the next call arrival time which is treated differently by the two systems, say at time t_β .

Let

$$p_{k0}(t) = P\{k\text{th call admitted to system 0 will be active at } T \mid \text{it is active at } t\},$$

$$p_{k1}(t) = P\{k\text{th call admitted to system 1 will be active at } T \mid \text{it is active at } t\},$$

$$P_{D0}(t) = P_D(t \mid \vec{s}_0(t)) = t\text{-dropping probability of a call admitted at } t \text{ in system 0,}$$

$$P_{D1}(t) = P_D(t \mid \vec{s}_1(t)) = t\text{-dropping probability of a call admitted at } t \text{ in system 1.}$$

Clearly, if admitted call k has completed service in system X by time t , then $p_{kX}(t) = 0$. Since the call holding time distributions have increasing hazard rate functions, we have $p_{k0}(t_s^+) \leq p_{k1}(t_s^+) \forall$ calls k active in either system. Moreover, this inequality remains true for all time t at least until t_β . Since $p_{k0}(t_\beta^-) \leq p_{k1}(t_\beta^-)$ for all calls k active in system 0, and since the dropping probability depends on call ages only through computation of the $p_{kX}(t)$'s, we have $P_{D0}(t_\beta) \leq P_{D1}(t_\beta)$. Thus, as before, the only possible way for the two policies to treat the call request differently is for the CVIFR policy in system 0 to accept the call, and the other conforming policy in system 1 to reject it.

The same arguments hold whenever semi-equality or equality is achieved between the two systems. Thus, there is never any time that system 1 admits more calls than system 0, and so the theorem holds. \square

8. Conclusions

Several existing and emerging networks, such as LEOS, mobile satellite personal communications networks, and multipriority reservation networks, have capacities which vary over time for some or all types of calls carried by these networks. With each capacity decrease having the potential to result in dropped calls, it is reasonable to include stochastic call dropping performance as part of the QOS guaranteed by the

network to accepted calls. It then becomes one of the duties of the CAC policy to ensure that these call dropping guarantees are met. The naive use of existing CAC policies which base admittance decisions using only the currently available capacity may lead to intolerable call dropping in capacity-varying networks, and thus novel CAC policies are required for these networks.

In this paper, we investigate using knowledge of future capacity changes, such as are available in NGOS satellite networks by their cyclic nature, to trade off some additional call blocking in order to meet any desired call dropping guarantee. We begin with the CVCAC policy for calls with exponentially distributed call holding times under an analytically simpler, yet still practical, dropping policy (LCFD). The general structure of the optimal CAC policy under this dropping policy provided insight into the form of the Admission Limit Curve (ALC), which bounds the conditions under which calls can be admitted under any causal, conforming pair of CAC and dropping policies. The ALC forms the basis of a CAC policy \mathcal{A} which, although nonconforming, is useful nonetheless because its blocking performance represents a tight lower bound on the blocking performance achievable by conforming CAC policies. The ALC also guides development of CAC policies such as CVGH, which seek to minimize call blocking by maximizing the area in the Admissible Region under which the CAC does admit calls.

The CVCAC, \mathcal{A} , and CVGH CAC policies presented in this paper represent the above progression of CAC policies for calls with exponentially distributed call holding times. The CVIFR CAC policy also presented here represents the first step in extending this progression to calls with IFR call holding times. Work is ongoing on the second step of the progression, namely, the investigation of the conditions which limit admissibility of calls with IFR-distributed call holding times.

The work presented here assumes that future capacity changes are deterministic. Research is also ongoing to determine constraints and to develop CAC policies for networks with stochastic knowledge of future capacity changes, both in the capacity change time and in the future capacity level.

References

- [1] M. Degermark, T. Kohler, S. Pink and O. Schelen, Advance reservations for predictive service in the Internet, *Multimedia Systems* 5(3) (May 1997) 177–186.
- [2] D. Ferrari, A. Gupta and G. Ventre, Distributed advance reservation of real-time connections, *Multimedia Systems* 5(3) (May 1997) 187–198.
- [3] D. Levine, I. Akyildiz and M. Naghshineh, A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept, *IEEE/ACM Transactions on Networking* 5(1) (February 1997) 1–12.
- [4] Z. Liu and M. El Zarki, SIR-based call admission control for DS-CDMA cellular systems, *IEEE Journal on Selected Areas in Communications* 12(4) (May 1994) 638–644.
- [5] M. Naghshineh and M. Schwartz, Distributed call admission control in mobile/wireless networks, *IEEE Journal on Selected Areas in Communications* 14(4) (May 1996) 711–717.
- [6] H. Perros and K. Elsayed, Call admission control schemes: A review, *IEEE Communications Magazine* 34(11) (November 1996) 82–91.

- [7] R. Ramjee, R. Nagarajan and D. Towsley, On optimal call admission control in cellular networks, in: *Proc. of INFOCOM '96*, Vol. 1, pp. 43–50.
- [8] I. Rubin and S. Shambayati, Performance evaluation of a reservation random access scheme for packetized wireless systems with call control and hand-off loading, *Wireless Networks* 1(2) 147–160.
- [9] J. Siwko and I. Rubin, Call admission control for mobile satellite communication networks, in: *Proc. of ICII '98*, pp. 432–435.
- [10] J. Siwko and I. Rubin, Call admission control for non-geostationary orbit satellite networks and other capacity-varying networks, in: *Proc. of SPECTS '99*.
- [11] J. Siwko and I. Rubin, Call admission control for non-geostationary orbit satellite networks and other capacity-varying networks, *International Journal of Satellite Communications* 18(2) (March 2000) 87–106.
- [12] H. Uzunalioglu, J. Evans and J. Gowens, A connection admission control algorithm for low earth orbit satellite networks, in: *Proc. of ICC '99*.
- [13] D. Wischik and A. Greenberg, Admission control for booking ahead shared resources, in: *Proc. of INFOCOM '98*, Vol. 2, pp. 873–882.
- [14] W. Wu, E. Miller, W. Pritchard and R. Pickholtz, Mobile satellite communications, *IEEE Proceedings* 82(9) (September 1994) 1431–1448.