# Real-time Prioritized Call Admission Control in a Base Station Scheduler *

Jay R. Moorman
jrmoorma@uiuc.edu
University of Illinois

John W. Lockwood
lockwood@arl.wustl.edu
Washington University, St. Louis

Sung-Mo Kang
kang@ece.uiuc.edu
University of Illinois

**Abstract**

*With the deployment of packetized wireless networks, the need for Quality of Service is becoming increasingly important. In order for QoS to be implemented and efficiently supported, a number of key areas in wired/wireless integration need to be addressed. One key issue is the call admission policy for the base station scheduler. In this paper we provide a call admission control policy that is designed to make efficient use of the limited wireless channel. This scheme takes advantage of the diverse traffic characteristics in order to admit flows based on the requested traffic parameters. Simulation results are included to verify the validity of our approach and its importance in future wireless communication.*

## 1 Introduction

In the next generation of networks we will begin to see the true convergence of voice, multimedia, and data traffic. This merging of various dedicated networks will occur both in the wired and wireless arenas. In order to support such a wide range of traffic on a network, the infrastructure must be capable of coping with varying QoS requirements. This support includes both the call admission and the subsequent scheduling of packet transmissions.

In this paper we focus on the call admission control policy necessary to provide the QoS guarantees in a base station scheduler. Much work in this area has focused on cellular voice networks with predominantly constant bit-rate voice traffic. Very little work has been presented regarding packet networks with very different types of traffic on a wireless link.

The contribution of this paper is the development and analysis of a control policy for admitting flows into a base station scheduler where each flow has different traffic requirements. In our previous work, we presented a scheduling algorithm designed to fit into a broader framework such as in the scheduling multiplexer and QoS sublayer of [1]. This scheduler was shown to provide bounded traffic guarantees to multiple classes of traffic in [2]. This bound can only be guaranteed so long as a proper call admission algorithm is used to admit calls. It is proposed that a single broad call admission policy cannot be implemented for all traffic types, but rather must be specialized to treat the various traffic classes with individually tailored policies. The merits of this approach will be demonstrated and verified with an approximate model and simulations.

This paper is organized as follows. Section 2 looks at the objectives for call admission and previous work in the area. Section 3 describes the wireless network model and call admission assumptions. Section 4 provides the detailed algorithms for call admission of the various traffic classes. In Section 5 the analytical model and simulation results for the proposed policy are presented. The policy is also compared and contrasted to other possible admission policy candidates. Section 6 summarizes the results, and Section 7 outlines future work. Finally, Section 8 concludes the paper.

*This work is supported in part by a grant from Nortel Networks.

## 2 Call Admission Background

### 2.1 Call Admission Goals

The area of Call Admission Control (CAC) is of vital importance, particularly for true Quality of Service traffic guarantees. There are a number of important issues that must be defined and addressed when developing a call admission algorithm for any network. The goals of a CAC algorithm should satisfy a number of different constraints. The algorithm should maximize the use of the available bandwidth, either in calls accepted or traffic scheduled according to channel utilization. It should do this in an efficient and fair manner for all currently connected flows as well as incoming call requests. The blocking probability for new calls should be minimized. It should also minimize the call drop probability. This is the probability that a currently connected call will be dropped for a new incoming call. Reduction of service to currently connected flows for new calls should also be minimized. In particular, the dropping or reduction of currently connected calls should be eliminated for some types of traffic, while only minimized for lower priority traffic. The following summarizes the goals of our call admission policy in order of importance:

1. Maximize channel utilization in a fair manner to all flows.
2. Minimize the probability of dropping connected calls.
3. Minimize the reduction of service for connected calls.
4. Minimize the probability of blocking a new call.

The constraints upon which new calls are admitted into the system should be based on the ability of the scheduler to meet the requested Quality of Service parameters. This includes the necessary bandwidth requirements of a new flow, as well as any guaranteed delay requirements for real-time traffic. Additionally, the requested cell loss ratios should be factored into admission decisions. Finally, the CAC algorithm must only admit flows if sufficient buffer space is available in the base station.

In our approach, we develop a policy that will group the requesting calls into traffic classes in order to more highly specialize the algorithm. In particular, we perform this classification according to the traffic parameters such as CBR, rtVBR, nrtVBR, ABR, and UBR as specified in [3].

### 2.2 Comparable Work

The area of call admission control is not in the least a new topic to the networking world. However, most work in this area has not addressed wireless specific issues or has focused on only voice traffic. Our work is an attempt to make the most efficient use of the wireless channel for a wide range of traffic types and requirements. Some of the related wireless call admission work before us is presented below.

The work by Ho and Lea [4] attempts to maximize the normalized channel throughput when there are defined constraints on new call and handoff blocking probabilities. A number of different CAC policies are examined, and a linear programming policy is used for increased throughput. This work focuses on the handoff across cells in a more mobile network. The target network of our work is expected to have many less handoffs. Their results are also specific to a single type of wireless traffic such as voice calls. An exact model is used for the analysis which limits the scenario to a small number of simulated calls (tokens) across a few cells.

An adaptive algorithm for call admission of multimedia traffic under variable channel conditions is presented by Kwon et al. in [5]. A reallocation algorithm is used to adaptively redistribute bandwidth. This can cause all currently connected traffic to receive reduced service. In our algorithm, some traffic may receive reduced service but only if it is of a low priority nature, can handle increased delay, and was conditionally admitted with knowledge that it may receive reduced service. The notion of QoS supported in this work is based on the number of calls below a target bandwidth. Our notion of QoS is to guarantee to each individual flow the requested level of service. The presented results are again focused on a single class of traffic with identical characteristics.

Kakani et al. present a wireless network CAC framework in [6] that supports QoS requirements for two different levels of traffic. This work focuses mostly on the scheduling of traffic in order to take advantage of allocated bandwidth that individual flows are not using. The throughput of the system is allowed to degrade so that the overall performance can be increased with less idle time. The defined QoS parameter is based on the throughput and number of users in the system. This is in contrast to our notion of QoS based on the requested service level of each flow.

A hybrid cutoff priority scheme for call admission is proposed in [7] by Li et al. Here multiple classes of traffic are considered where each level has its own cutoff threshold at which point only handoff calls are accepted and new calls are blocked. In addition, calls are allowed to be queued if channel capacity is currently unavailable. Our work is similar in that it separates calls according to the traffic type, however, we have focused less on the ability to accept handoff calls. Our algorithms are designed to provide a base station with a CAC policy that will allow it to make the best use of the channel bandwidth such that it can schedule packets to meet guaranteed delays and cell rates. Our algorithm also does not allow incoming calls to be queued as we do not expect applications to accept long queuing delays waiting for access. Instead, another available wireless cell should be contacted. If a call cannot be serviced it will be denied instead of queued for later service.

Finally, in [8] a reservation scheme is used by Rubin and Shambayati with a call admission policy that achieves a throughput capacity greater than a system with no CAC policy. The main thrust of this work focuses on support of handoff calls to guarantee these calls a reduced level of blocking. All traffic is assumed to be voice calls.

Despite the differences in the focus of much of this work, many of the applied methods can be used in our situation. However, we will show that for diverse traffic in a wireless system it is best to apply traffic specific admission requirements in order to best utilize the channel capacity.

## 3 Wireless Network Model

In this paper we assume a high-speed wired backbone that is extended to a packetized wireless cellular network. The wireless cellular network is divided into partially overlapping cells that transmit on different logical channels. Each cell contains a base station that is connected to the wired network. The base station schedules packets for transmission in its particular cell for both the uplink (mobile host to base station) and the downlink (base station to mobile host) channel.

The channel is said to be in error for a particular flow if the base station cannot communicate with the particular mobile using that flow. Errors are assumed to be location dependent since one mobile experiencing a fade will not affect the transmission of another mobile at a different location. Errors are also assumed to be typically bursty in the wireless environment.

QoS is supported by meeting bandwidth and delay parameters that are negotiated at call admission. These parameters, such as cell rate or maximum delay, are then guaranteed by the network until the connection is terminated.

The call admission criteria is highly dependent on the rate of incoming and outgoing calls. For each particular traffic class there are a number of model parameters that must be considered. These include the rate of incoming calls in handoff from another cell, the rate of outgoing handoff calls, the rate of new calls generated within the cell, and the rate of termination of calls in the cell. For the rest of the paper, for each traffic class a single incoming call rate ($\lambda_k$), and a single outgoing call rate ($\mu_k$) have been used to include both handoff calls and new/terminated calls within the wireless cell. Once a CAC policy has been defined the handoff calls can be separated and treated differently if desired. In Table 1, the notation is specified for the rest of the paper.

## 4 Proposal

### 4.1 Call Admission Policies

Call admission policies can be categorized using the notation of Table 1. The policies described below have been based on policies from literature with additional manipulations to fit

| Notation | Definition |
|---|---|
| $P_b$ | The probability of a call being blocked. |
| $P_d$ | The probability of a call being dropped. |
| $\lambda_k$ | The arrival rate of calls/flows in class k. |
| $1/\mu_k$ | The mean length of calls in class k. |
| $L_p$ | The length of packets (cells) in all flows. |
| $C$ | Channel Capacity. |
| $\zeta$ | Total number of connected calls. |
| $W$ | Window size. |
| $\alpha$ | Dynamic window parameter |
| $\gamma_i$ | Termination parameter for flow i. |
| $R_i$ | The rate for flow i. |

Table 1: Paper Notation

into these general admission policy categories. The descriptions are given in terms of the ability of the scheduler to meet bandwidth requirements for arriving calls. The goal of each admission policy is still to satisfy the admission criteria discussed in Section 2.1

In order to admit flows according to any CAC policy the bandwidth of the call must be known. This is typically done during the negotiation phase when the caller requests a certain level of service. The schedule is thus privy to the parameters of Peak Cell Rate (PCR), Sustained Cell Rate (SCR), and Minimum Cell Rate (MCR). Our initial policies use these parameters as the requested capacity for a particular traffic flow. However, work has been done regarding the approximation of effective capacity on a wireline link in [9]. This approach can be applied to the wireless channel in a similar manner to better estimate the actual capacity needed by the requesting flow. This method, based on large deviation theory, provides an exponential approximation that can take advantage of statistical multiplexing while ensuring no constraints on the Cell Loss Ratio (CLR) are violated.

#### 4.1.1 *Policy 1*: Equal Access Sharing

In this CAC policy, no single flow or traffic class is treated any differently than any other flow or traffic class. This is the simplest scheme and involves checking to guarantee that the requested bandwidth is available for the new call. If the bandwidth requirements can be met, the flow is admitted, assigned a weight by the base station scheduler as shown in [2], and allocated transmission slots as needed.

When viewed in context of the original goals presented in Section 2.1, we find the following:

1. First Come First Serve (FCFS) channel access for all flows.
2. $P_d = 0$ (Policy 1 will never force calls to drop.)
3. No connected call will receive reduced service.
4. $P_b \approx 0$ with underutilized channel.
   $P_b \approx 1$ once channel is full.

#### 4.1.2 *Policy 2*: Equal Access Sharing with Reserve

For this CAC policy all traffic is admitted equally within a specified bandwidth of the maximum channel capacity. Once the available channel capacity has been used, only calls that are of a high priority will be admitted to use the remaining reserved bandwidth. This has the effect of prioritizing a small number of calls above the other traffic classes. However, this method will suffer from the inability to admit as many flows as other schemes under normal load. Note that channel bandwidth will not normally be wasted under the Multiclass Priority Fair Queuing (MPFQ) scheduler[2] unless there are no backlogged flows needing to send traffic.

In context of the CAC goals:

1. FCFS channel access until emergency bandwidth reserve is reached.
2. $P_d = 0$ (Policy 2 will never force calls to drop.)
3. No connected call will receive reduced service.
4. $P_b \approx 0$ with underutilized channel.
   $P_b \approx 1$ for all non-emergency calls once all but emergency reserve is used.

#### 4.1.3 *Policy 3*: Equal Access Sharing with Priority

Under this CAC policy traffic will be admitted equally according to the ability of the scheduler to satisfy the bandwidth constraints. However, in the original equal sharing scenario, once the channel was completely allocated, all calls would be blocked until current traffic was reduced. In this situation, an incoming priority call (emergency, handoff, voice, real-time, or higher paying customers) can be admitted, even though the current capacity does not support it. The scheduler will then look to drop a call of lower priority non-real time traffic. This allows the channel allocation to be maximized for high priority traffic. In effect, the higher priority traffic will be given bandwidth at the expense of lower traffic classes which may be dropped by the scheduler.

In context of the CAC goals:

1. FCFS channel access until channel is fully allocated. Then prioritized access.
2. $P_d = 0$ for high priority traffic. When prioritized admission occurs, $P_d$ increases for low priority traffic.
3. Low priority call (nrt) may be completely dropped. No calls will remain connected and receive reduced service.
4. $P_b = 0$ for high priority traffic.
   $P_b$ is low but increasing for low priority traffic.

#### 4.1.4 *Policy 4*: Threshold Access Sharing

In this CAC policy three levels of call admission are enabled. At the lowest level, when the wireless channel is highly underutilized, all incoming calls that can be satisfied are admitted. After a certain threshold ($T_{low}$), when channel allocation is becoming more highly utilized, all calls are still admitted, but lower priority calls are admitted conditionally. This means these calls are aware that the channel is heavily used, and know that they run the possibility of losing their connection or having their service reduced if more traffic arrives. Exceeding the upper threshold ($T_{high}$) puts the CAC policy into a state of prioritized admission. The lower priority traffic will then receive reduced service. This reduced service will be targeted at the conditionally admitted flows. In effect, the higher priority traffic will be given limited bandwidth at the expense of additional delay in the lower traffic classes. The channel will be in a state of complete utilization, or even over-utilization. Only new high priority calls will be admitted, while lower priority calls already admitted will receive reduced service. A diagram of this CAC policy is shown in Figure 1.

In context of the CAC goals:

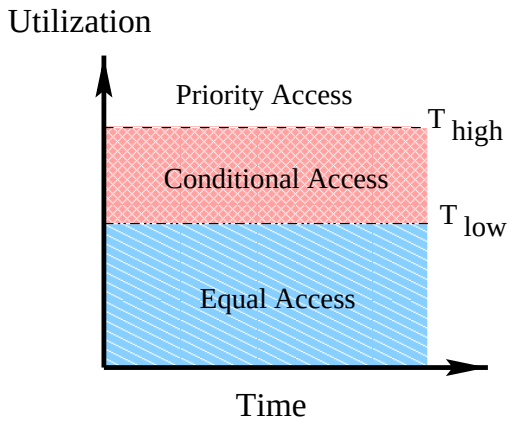



Figure 1: Policy 4: Threshold Access Sharing

1. FCFS channel access while: $(Utilization \leq T_{low})$.
   Conditional access while: $(T_{low} < Utilization < T_{high})$.
   Prioritized access while: $(Utilization \geq T_{High})$.
2. $P_d = 0$. (Policy 4 does not currently drop any calls even when $(T_{high})$ has been exceeded.)
3. Low priority calls (nrt) may receive reduced service if they were conditionally admitted.
4. $P_b = 0$ for high priority traffic.
   $P_b$ increases towards 1 for low priority traffic.

### 4.2 Buffer Requirements

Internet traffic is inherently bursty. For a base station serving hundreds of mobile users, it is important to consider allocation of memory in the base station. A key component of the call admission policy is the guarantee that enough buffer space will be available to handle the agreed upon traffic levels. This becomes particularly vital as the number of traffic flows requiring buffer space increases significantly.

Analysis was performed on the MPFQ scheduling algorithm in [2] to determine the necessary buffer requirements. These results provided the minimum buffering needed to fully support all incoming traffic requests. Any particular flow must be capable of buffering packets to support its requested level of service. Due to the prioritized nature of the MPFQ algorithm, each flow must additionally be capable of buffering packets while the queues from the higher priority levels drain. This increases the buffer requirements as the priority level of the flow decreases.

If at any time, the call admission algorithm does not have enough buffer space to support a new call, it must be forced to deny admission to that request. The derived buffer requirements do not include additional space needed due to jitter in the network. However, with a small adjustment in buffer space to account for buffering during wireless error, the effect of jitter on a flow will also be covered.

### 4.3 Channel Knowledge

In addition to the call admission algorithm used to admit traffic, the ability of the CAC policy to correctly know the channel capacity is of utmost importance. For the wireless channel, the bandwidth may vary significantly over even small time intervals. If the base station is unaware of this variation it may well over-admit traffic into the scheduler, and thus agree on contracts that it is incapable of fulfilling. To avoid this scenario, there are two methods that can be used to properly admit flows into the schedule. The first is a method to estimate channel utilization. In other words, to be able to measure how much data is currently being sent and/or received. The second method involves the ability of the base station to determine when a call has been terminated. This is especially difficult under the assumptions of mobility and channel error where a roaming connection may not have an explicit tear down interaction with the base station.

#### 4.3.1 Bandwidth Estimates

In order to admit new calls with guarantees, the base station must have an approximation to the amount of traffic it is capable of sending. In our scheme, this is done by traffic estimates over time.

A dynamic window is used to calculate a moving average of the channel bandwidth. The size of the window is based on a dimensionless parameter ($\alpha$) which allows the estimate size to grow or shrink over time. This parameter depends on the rate of incoming/outgoing calls. As more fluctuation in the schedule occurs, the time window will shrink; forcing more instantaneous traffic estimates to follow the dynamics of the admission policy. As the flows in the system stabilize to less active changes, the estimate will grow to encompass a larger time window and a more average traffic estimate, with less overhead involved. The window size is limited to a maximum size ($W_{max}$) such that the average estimate does not become too decoupled from the actual traffic being sent.

$$W_i = min\{W_{max}, \alpha(W_{i-1} + 1)\} \tag{1}$$

$$\alpha = \left( \frac{\zeta}{\zeta + \left(\sum_k \lambda_k + \sum_k \mu_k\right) \Delta t} \right) \tag{2}$$

The moving window bandwidth estimates can be calculated from Equation 1 where $\alpha$ is found from Equation 2. This value of $\alpha$ is based on the volatility in arriving and departing calls over the time period ($\Delta t$) since the window was last computed. Thus the system need only keep track of the total number of connections that have been admitted ($\sum_k \lambda_k \cdot \Delta t$), the total number of connections that have been terminated ($\sum_k \mu_k \cdot \Delta t$), and the total number of connected calls ($\zeta$). As more activity occurs, $\alpha$ will asymptotically decrease towards 0, while no activity will result in $\alpha = 1$. In Equation 1 the maximum window size is limited. If the window is below $W_{max}$, it will either be fractionally reduced by $\alpha$ in the case of traffic changes, or will increase linearly in the case of no changes.

The graph in Figure 2 shows how $\alpha$ varies as the traffic intensity increases. Each curve shown is modeled with a different number of connected calls. The variation in incoming and outgoing calls is shown on the x-axis where $\lambda_k = \mu_k$ in this scenario. As the traffic intensity increases, the ratio $\alpha$ falls off exponentially towards 0. The rate of decrease is faster when the total number of calls is smaller. This is based on the assumption that traffic changes in a network with a small number of calls will cause greater overall turbulence to the system then changes in a heavily loaded network with a large number of calls.

The dynamic window approach has been modeled to produce an exponential decrease and a linear increase, similar to TCP. However, instead of adjusting the window based on

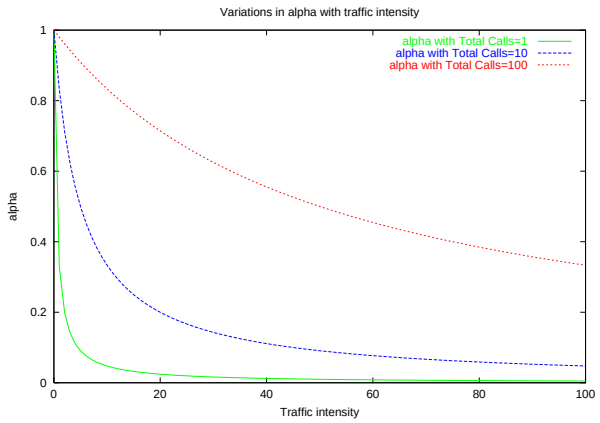



Figure 2: Variation in parameter alpha

congestion and timeouts, the window size is determined by the incoming and outgoing calls.

#### 4.3.2 Terminated Call Estimates

The ability of the base station to properly admit new calls also will be highly dependent on the actual traffic flows that are included in the schedule. Even when these flows fall in error, and are not using channel bandwidth, the scheduler must anticipate the end of their fade and the renewed connection with the data traffic. However, in scenarios where these connections fail completely, the base station must be able to make a decision that the call has been terminated.

For our CAC policy, we perform this by weighting each flow with a connection parameter ($\gamma$) between 0 and 1. Each flow is initially connected with a value of $\gamma = 1$, and will remain at that level as long as communication is established. However, as soon as the channel experiences error, the parameter will be allowed to fade towards 0. With each scheduled slot that remains in error for that flow, the parameter will be adjusted to a reduced level. Once the parameter reaches 0, the call will be considered to be terminated and will no longer be included in the schedule. This enables the base station to then recapture that channel capacity, and allow more incoming calls to gain channel access. If at any point in the fading process the call regains connectivity, the parameter is reinitialized to 1 as a fully connected mobile.

$$\gamma_i = \frac{4\sigma_z - \frac{\eta}{R_i}}{4\sigma_z} \tag{3}$$

The termination parameter calculated in Equation 3 is based on the distribution of a long-term fade, as well as the total time of the fade. The total number of missed slots is kept track of via $\eta$. When this term is divided by the flows rate ($R_i$), the time of the current fade can be found. The density function of a long-term fade (Equation 4), is based on the log-normal distribution[10]. Instead of an exact error model to use in determining the termination estimation, measurements might also be taken and a distribution calculated. The best error model to use has been investigated elsewhere in [11], but still remains an open research problem.

$$f_z(m) = \frac{1}{m\sigma_m\sqrt{2\pi}} e^{\left[-(\log m - \overline{m})^2/(2\sigma_m^2)\right]}, m > 0$$

$\overline{m}$ = mean of $\log m$

$\sigma_m$ = standard deviation of $\log m$ (4)

Once the fade time has exceeded a bounded section of the fade distribution, the connection can be considered to be lost. In order to find this point, the density function in Equation 4 can be used to determine both the mean ($\mu_z$) and variance ($\sigma_z^2$). These values for the log-normal distribution are shown in Equation 5 and Equation 6 respectively. The variance can then be used to determine the standard deviation ($\sigma_z$) of the distribution.

$$\mu_z = e^{\overline{m}+\sigma_m^2/2} \tag{5}$$

$$\sigma_z^2 = e^{2\overline{m}+\sigma_m^2}(e^{\sigma_m^2} - 1) \tag{6}$$

The standard deviation $\sigma_z$ is used to find a lost call. When the time of the current fade is tracked, it will exceed $4\sigma_z$ with such small probability that this value was chosen as the cutoff termination point. Once the fade has taken this long, the call is assumed to be completely lost to the base station communication. Thus once $\gamma_i \leq 0$ for a particular flow $i$, it will no longer be considered for the channel schedule by the base station.

This method of terminated estimation shifts the wireline paradigm of virtual circuits slightly. Since the channel is wireless, the virtual circuit cannot always be guaranteed to remain connected and to satisfy its initial traffic request. The mobile may also not be able to tear down a specific connection. The estimation of terminated calls allows this bandwidth to be reclaimed. It also fits in well with an IP based network without the notion of virtual circuits. There still needs to be a call admission phase, but the scheduler can judge the continued connectivity of a traffic flow (IP or ATM) according to its own estimation.

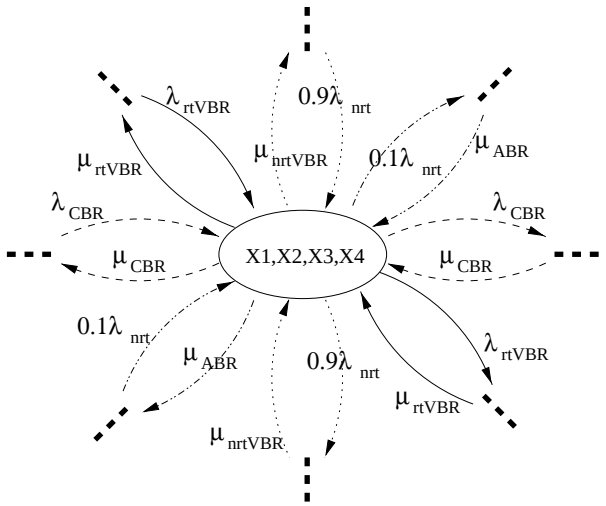
## 5 Comparison

This section looks at the different call admission policies using simulation and a Markov model representation. All comparisons were evaluated using stochastic activity networks modeled in the *Ultra*SAN [12] software package [1]

### 5.1 Call Admission Model

To evaluate the various call admission policies, an approximate model was developed that included all of the necessary components of a general system. This model was then specialized with the necessary rules to apply each particular call admission algorithm. The basic idea is to separate the incoming and outgoing traffic into the prioritized traffic levels. At each level the individual call admission rules can be applied. Simulations can then be performed on each of the variations to compare and contrast the different algorithms.

The model used to simulate and analyze the various CAC algorithms is based on a continuous-time Markov chain. Every state is tagged with four markings ($X1, X2, X3, X4$), each of which specifies the number of tokens for a particular type of traffic. A single state of the entire state-transition diagram is shown in Figure 3. The transition into the state can occur either due to a generation/handoff of a new incoming call ($\lambda_k$), or due to the termination/handoff of a completed call ($\mu_k$). In either case, the value of $k$ specifies the particular traffic class. The Markov model assumes all transitions to be exponentially distributed. Note that




X1: CBR Tokens
X2: rtVBR Tokens
X3: nrtVBR Tokens
X4: ABR Tokens

Figure 3: Markov Model

the non-real time traffic has a single generation rate ($\lambda_{nrt}$) which is divided into the nrtVBR and ABR traffic. This was chosen according the traffic scenario in Section 5.2.

Once this approximate model is defined, various steady state parameters can be calculated such as $P_b$, $P_d$, or $P_{full}$. The probability the system will be completely filled to capacity (entire bandwidth has been completely allocated) can be seen in Equation 7. This consists of computing the sum of the probabilities of being in each state with a full system capacity $C$, where $x_1 + x_2 + x_3 + x_4 = C$. For a large model, this translates into performing a tremendous number of calculations, as well as computing the probabilities of being in each state with a very large matrix. To simplify this process, a SAN simulator with the necessary capabilities was utilized.

$$P_{full} = \sum_{i=1}^{C} \sum_{j=1}^{C-i} \sum_{k=1}^{C-j} \sum_{l=1}^{C-k} P(x_1 = i, x_2 = j, x_3 = k, x_4 = l) \quad (7)$$

The Markov model in Figure 3 was translated into a SAN model shown in Figure 4 for the *Ultra*SAN software package. At each traffic class level, two timed activities were used for both the intensity of the arrival process, and the intensity of the service time. Each level also contains a place to hold the number of tokens signifying the bandwidth consumed by that level. Since calls at each level are assumed to use different amounts of bandwidth, input (*k_size*) and output (*k_done*) gates are used to add or subtract the proper number of tokens corresponding to individual calls. The number of tokens used at each level is described in more detail in Section 5.2 Finally, an input gate (*capacityX*) is used before each input activity in order to apply the necessary CAC blocking, dropping, or admission policy.

The basic SAN structure shown in Figure 4 was used for each of the CAC policies. The interesting variation occurred in the *capacityX* input gates. The general approach to each policy presented in the table is summarized below:

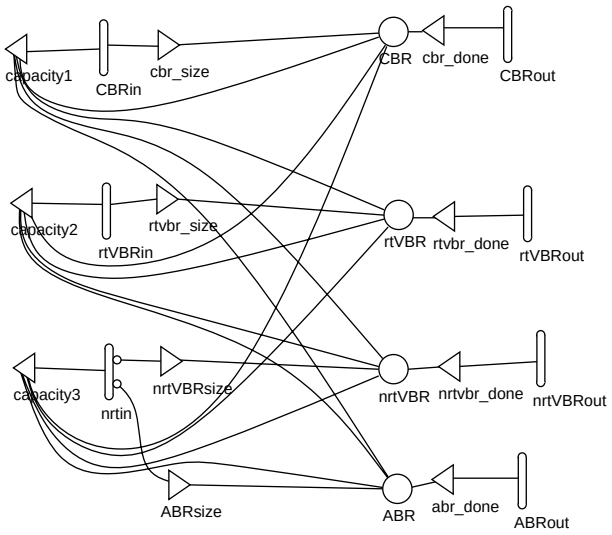



Figure 4: SAN Model

**Policy 1:** This policy is simply checking at each level to verify that enough capacity exists in the system to support a single additional call of the respective traffic type at the required bandwidth level.

**Policy 2:** This policy is performing the same check as policy 1, with the added constraint that the overall bandwidth has been reduced by $Ereserve$.

**Policy 3:** This policy will only limit the CBR traffic if this high priority traffic has already completely consumed all available bandwidth. The remaining traffic levels perform the same bandwidth available check as that of policy 1. When additional CBR traffic is admitted, such that the total bandwidth allocated exceeds the available capacity, calls from the lowest priority class will begin to be dropped. The algorithm modeling this decision is described in the following pseudo-code:

```
if (Total bandwidth used > T_high)
/* Max Capacity Exceeded */
   if ( Any ABR Flows ) /* Drop ABR First */
      { Drop ABR flow and reclaim bandwidth }
   elsif ( Any nrtVBR Flows )
      /* Drop nrtVBR Flows second */
      { Drop nrtVBR flow and reclaim bandwidth }
   elsif ( Any rtVBR Flows )
      /* Drop rtVBR as last resort */
      { Drop rtVBR flow }
   endif
endif
```

**Policy 4:** This policy will allow CBR traffic until the available bandwidth has reached the high threshold $T_{high}$. Below this threshold the lower priority traffic will be admitted if bandwidth is available to support the requested call size. Once CBR traffic has been admitted above $T_{high}$, the conditionally admitted lower priority traffic will begin to receive reduced service.

### 5.2 Call Admission Simulations

Simulations were completed with varying input rates for each traffic class in order to evaluate the performance of each

admission policy. This evaluation of the four algorithms was accomplished using a steady state simulator. The incoming traffic intensities and service times were chosen to approximate a typically wireless LAN scenario. All simulations were run with a 95% confidence level.

The initial traffic scenario chosen is shown in Table 2. The total capacity of the system was assumed to be a 11 Mbit/sec wireless channel. This was approximated with $Capacity = 175$. The CBR traffic was simulated assuming 64 kbit voice calls arriving at a moderate rate and only lasting for a short duration. The rtVBR calls were assumed to be much larger video calls that arrived much less frequently. This traffic was also not assumed to stay in the system for long durations, though it was expected these would be longer transfers then voice calls. The incoming rate for non-real time traffic was assumed to be generated fairly quickly at 1/sec. This is based on the fact that most traffic at this level will be web based connections that happen very often. It has been found elsewhere that approximately 90% of internet TCP traffic is less than 10 packets (primarily due to web traffic), and 10% is much larger (such as for file transfers).[2] This is the basis for the division into nrtVBR and ABR flows. The nrtVBR flows are modeled as small size packets of very short duration. The ABR flows are also assumed to need a small amount of bandwidth for their MCR, but are expected to last for a much longer period of time.

In the simulations, the parameters specified in Table 2 were kept constant while a single input arrival rate was varied to determine the effect on the CAC policy. The CBR rate ($\lambda_{CBR}$) was simulated from 0.01 to 0.29. The rtVBR rate ($\lambda_{rtVBR}$) was simulated from 0.001 0.025. The nrt rate ($\lambda_{nrt}$) was simulated from 0.1 to 1.9. The value of $T_{low}$ was chosen to be approximately 50% of the channel capacity, while the value of $T_{high}$ was chosen to be the maximum 100% channel capacity. The graphs of these simulations are presented and discussed below.

### 5.2.1 Utilization

One of the most important goals in the call admission policy should be the channel utilization that can be achieved. In Figure 5 and Figure 6 the normalized utilizations are given as the arrival rates varied. From Figure 5 it can be seen that both policy 1 and policy 3 have nearly 100% channel utilization. The utilization of policy 2 suffers due to the reserved bandwidth. The reserved amount was chosen to support 10 calls of size 64 kbit each. This is approximately 6% of the channel capacity which can be seen by the 94% utilization level. The best channel utilization occurs with policy 4 since the channel can actually be over-utilized. This over-utilization is only slightly over 100% until the incoming CBR rate exceeds the CBR service rate at 0.05 and the system becomes unstable. The number of CBR calls will then accumulate as they are not being serviced as quickly as they are arriving. Recall that when the channel is being over-utilized the lower priority traffic will suffer through additional delay and reduced service.

The graph in Figure 6 shows the utilization as the incoming non-real time traffic rate varies. Here again it can be seen that policy 1 and policy 3 use nearly all of the channel capacity at 99%. When the channel is slightly underutilized, policy 3 performs slightly better then policy 1. This occurs since some of the higher bandwidth lower priority calls will be dropped leaving room for more higher priority CBR calls. An approximate 6% reduction in utilization is again

[2]Presented in a talk by Fred Baker, 1999

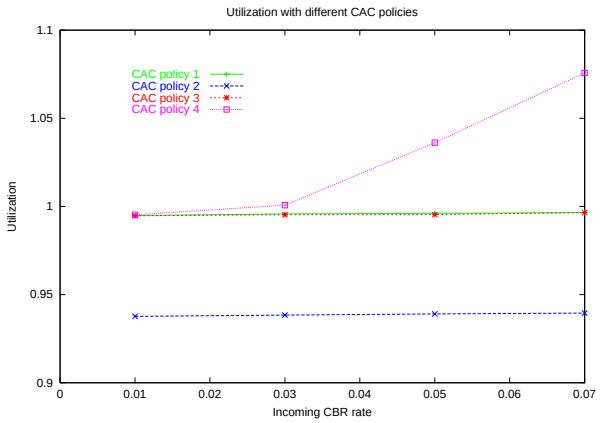



Figure 5: Utilization vs. $\lambda_{CBR}$

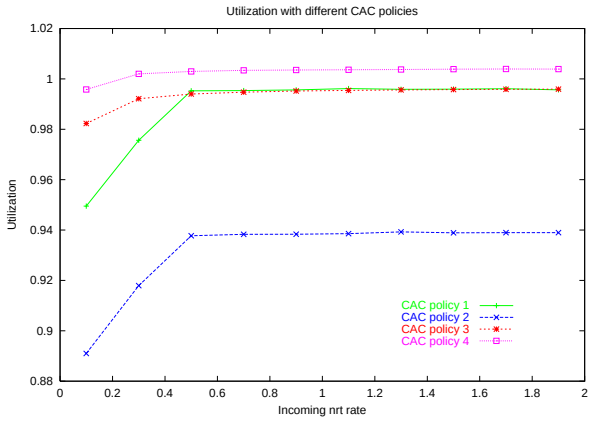



Figure 6: Utilization vs. $\lambda_{nrt}$

observed for policy 2. Finally, policy 4 again outperforms the other algorithms by allowing a utilization just over the maximum capacity.

### 5.2.2 Probability of Blocking

The goal of minimizing the probability of blocking new calls is also very important for any CAC algorithm. In Figures 7 and 8 the CBR blocking probabilities are shown. The graphs of Figures 9 and 10 show the non-real time blocking probabilities.

The graph shown in Figure 7 is the probability that a new CBR call will be blocked as the rate of arriving CBR calls increases. Both policy 1 and policy 2 have approximately the same curve which approaches 1 as the number of CBR calls overwhelm the system. These policies are both much more likely to block an incoming CBR call then either policy 3 or policy 4. The latter two policies also have similar curves. Notice that $P_b \approx 0$ until the rate $\lambda_{CBR} > \mu_{CBR}$ at 0.05. The CBR traffic then tends to flood the system causing it to fill the channel capacity and in effect block itself.

The graph of Figure 8 is again the probability that a new CBR call will be blocked, however, in this case it tracks the increasing arrival rate of the non-real time traffic. Policy 1 and 2 again have a higher blocking probability then either of the other two policies. The differences in the curves are also not significant enough to declare a clear advantage for

| Traffic | Bandwidth | Tokens | Arrival Rate | $\lambda_k$ | Call Duration | $\mu_k$ |
|---|---|---|---|---|---|---|
| CBR | 64 kbit | 1 | 1/30 sec | 0.035 | 20 sec | 0.05 |
| rtVBR | 3 Mbit | 45 | 1/15 min | 0.001 | 1 min | 0.017 |
| nrtVBR | 128 kbit | 2 | (90%) 1/sec | .9 | 1 sec | 0.333 |
| ABR | 128 kbit | 2 | (10%) 1/sec | .1 | 5 min | 0.003 |

Table 2: Input studies

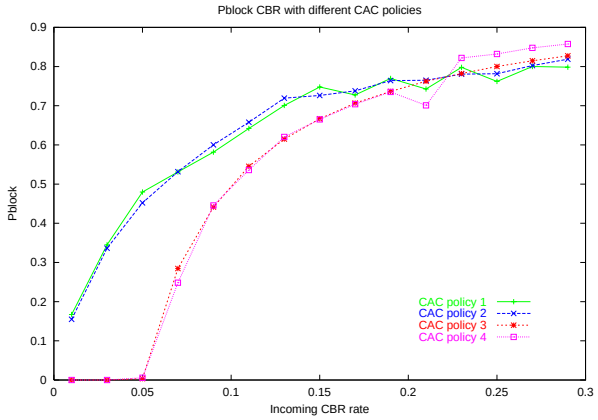



Figure 7: Probability CBR blocked vs. $\lambda_{CBR}$

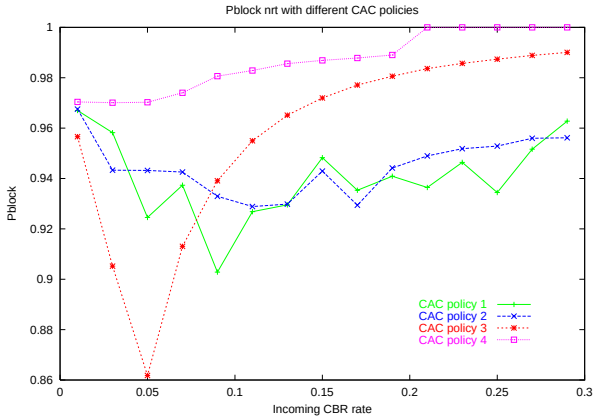



Figure 9: Probability nrt blocked vs. $\lambda_{CBR}$

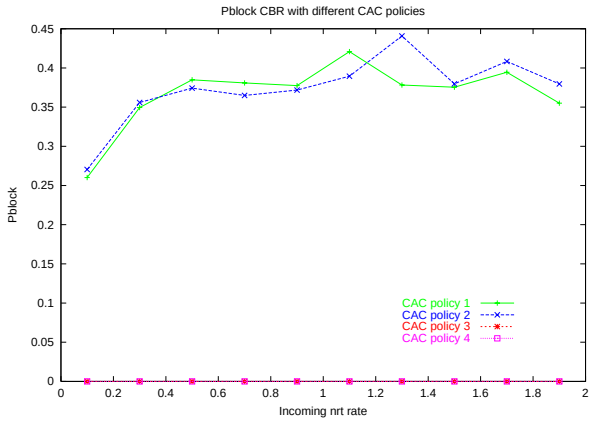



Figure 8: Probability CBR blocked vs. $\lambda_{nrt}$

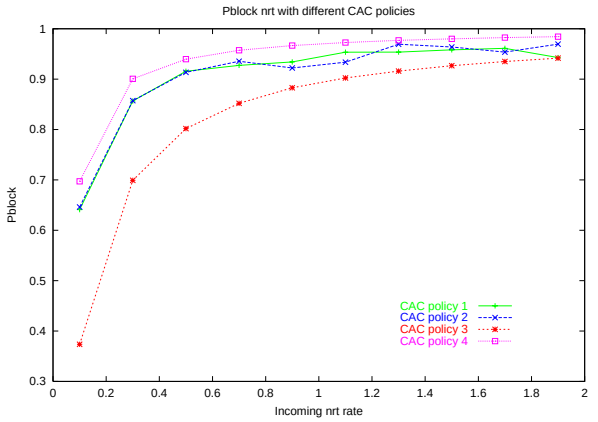



Figure 10: Probability nrt blocked vs. $\lambda_{nrt}$

either case. It can also be seen that both policy 3 and 4 have a $P_b = 0$ for the entire range. In both cases, this is due to the prioritized admission of CBR flows when channel capacity has been reached.

In Figure 9 the non-real time blocking probability is shown as the CBR arrival rate increases. In this case the $P_b$ for policy 4 is worse than any of the other graphs. Since the channel is allowed to be over-allocated it actually increases the time period that lower priority traffic can be blocked. This is the trade-off price for allowing prioritized CBR traffic into the schedule without dropping any currently connected calls. Both policy 1 and 2 demonstrate similar blocking probabilities for CBR traffic. The algorithm with the lowest $P_b$ is that of policy 3. This is a result of some traffic being dropped. When currently connected calls are forced to terminate, there is more bandwidth available for new incoming calls, thus reducing $P_b$. Notice that as $\lambda_{CBR}$ increases towards $\mu_{CBR}$ (0.05) the policy 3 blocking probability is actually decreasing. Since more CBR flows are coming into the system, more low priority traffic is being dropped thus decreasing the probability of blocking new calls from 0.96 to 0.86. However, once the service rate has been exceeded by the arrival rate, the CBR traffic begins filling up the bandwidth. This increases $P_b$ towards 1 for the non-real time traffic.

The final blocking graph is shown in Figure 10. This graph varies as the arrival rate $\lambda_{nrt}$ increases producing more nrtVBR and ABR traffic. The policy 4 traffic is again seen to have the worst blocking levels, while policy 3 maintains the clear advantage. Policy 1 and 2 are relatively similar to each other. Though the benefit seems to lie with policy 3 the effect from the dropping probability ($P_d$) must also be factored into the results.

### 5.2.3 Probability of Dropping and Reducing

Minimizing the probability of dropping currently connected calls is a key goal in our CAC evaluation. It is much better to deny the admission of new calls rather then drop a connected call with which a traffic contract has already been agreed upon. This is the key component leading to the policy 4 CAC algorithm. The reduction of service to lower priority traffic is a much more desirable outcome then the outright dropping of those calls. In Figures 11 and 12 the probability of dropping currently connected calls are shown for policy 3. No other policy will allow a connected call to be dropped by the CAC algorithm. The graphs also show the probability of a conditionally admitted low priority call receiving reduced service for policy 4.

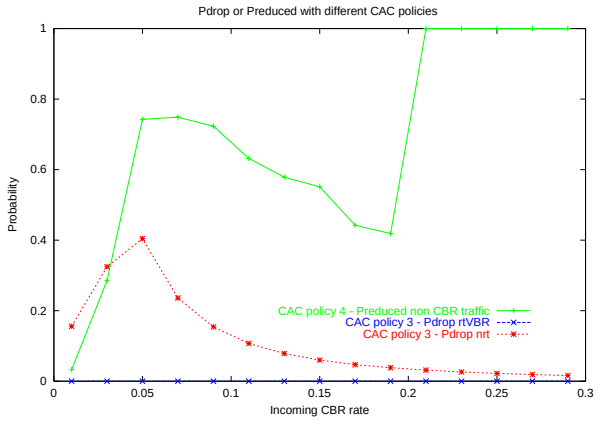



Figure 11: Probability nrt dropped/reduced vs. $\lambda_{CBR}$

In Figure 11, the probability of dropping a call, or of a call receiving reduced service are shown as the CBR arrival rate increases. In our simulation, the rtVBR traffic was never forced to drop any calls. However, this is predominantly due to the large bandwidth requirements needed for our traffic scenario. Since the scenario is designed to stress the channel utilization, the larger capacity rtVBR traffic was often blocked during admission. The non-real time traffic had an increasing drop rate as the CBR arrival rate approached its service rate. This probability peaked around 40%. Once the CBR arrival rate exceeded the service rate, the drop rate actually decreases since the probability of being blocked significantly increases (Figure 9). The probability of receiving reduced service also increases greatly as the arrival/service ratio approaches 1. This means that under policy 4 it is much more likely that lower priority traffic will receive reduced service then it would be dropped under policy 3. However, the actual values are much less important then the fact that under policy 4 no traffic will ever be forced to drop but will simply receive degraded service.

In Figure 12 the dropping/reduction probabilities are shown as the non-real time arrival rate increases. In this scenario, no rtVBR traffic is forced to be dropped. Once again it can be seen that the probability of receiving reduced service under policy 4 exceeds the probability of being dropped under policy 3. We feel the possibility of being dropped far outweighs the possible reduction in service through smaller bandwidth allocation and increased packet delays.

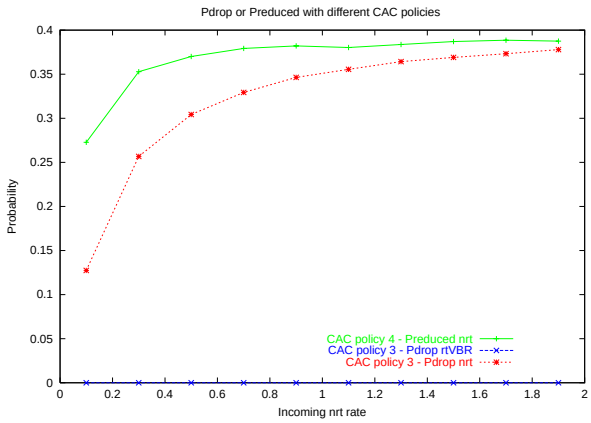



Figure 12: Probability nrt dropped/reduced vs. $\lambda_{nrt}$

## 6 Results

Careful consideration was necessary when deciding upon the best call admission policy. After viewing the policy comparisons, analyzing the simulation graphs, and evaluating the CAC algorithms in context of the original goals, some clear advantages were found with policy 4, threshold access sharing. However, choosing this algorithm has some tradeoffs that the network calls must be able to tolerate.

One of the biggest advantages to policy 4 is the clear improvement over the other three policies in network utilization. Since this satisfied the requirements of our primary goal, it weighted heavily in the comparison. This advantage stems from the fact that the channel is actually allowed to be over-utilized so that higher priority calls are admitted into the system. Lower priority calls will suffer in reduced service and increased delay. A capable scheduling algorithm, such as MPFQ [2] must be used in order to allow this over-utilization, and a bound must be enforced in a practical implementation.

The probability of blocking new calls was the second goal used in the comparisons. For high priority CBR traffic both the policy 3 and policy 4 algorithms provide $P_b = 0$. The lower priority traffic actually receives the worst $P_b$ from policy 4. This occurs due to the over-allocation of resources causing more new low priority calls to be denied. Policy 3 contains the lowest $P_b$ for non-real time calls. However, this must be taken in context of the third goal: the probability that calls will be dropped.

When comparing the policies according to the dropping probability $P_d$, no single policy can be chosen as the best, but only singled out as the worst. The only CAC algorithm that actually allows calls to be forced to terminate is policy 3. This weighs very heavily in not choosing this policy as the overall best call admission option.

The final goal used to evaluate the various policies was the probability of receiving reduced service. The only policy that allows this to occur is policy 4. This would appear to count as a mark against using threshold access sharing. However, when taken in context of the entire policy, it is not as dire as it might first appear. First, the reduction of service to high priority CBR traffic never occurs. Secondly, the reduction of service to low priority non-real time traffic only occurs when those flows have been admitted conditionally. Thus they are aware they may receive reduced service in the future. Since this traffic does not contain delay bounds

some additional network delay can be tolerated.

When all things are considered, policy 4 wins out as the best candidate for a CAC policy with QoS support. The advantages in utilization, blocking probability, and dropping probability far outweigh the tradeoff in reduced service to lower traffic classes. This is a key reason why a call admission policy that treats individual traffic classes uniquely is a very important step for ubiquitous wireless computing.

## 7 Future Work

While developing the call admission work in this paper we found a number of issues that warrant further study and research. Though these issues were not covered in the scope of this paper, they remain interesting and important areas for future research work.

The results of Section 6 lead to the choice of *Policy 4: Threshold Access Sharing* for our call admission policy. Though the benefits of this algorithm proved advantageous for the wireless prioritized traffic, the parameters involved in the policy were arbitrarily chosen for simulation. In other words, the threshold values were chosen as $T_{high} = Capacity$ and $T_{low} \simeq Capacity/2$. Future analysis will be performed to determine an approximate optimal value for these threshold levels.

A second issue needing further study involves the primary CAC goal. Our simulation results addressed the utilization of the channel but did not focus on the fairness of channel access. In particular, the threshold access sharing with CBR traffic receiving prioritized admission needs to be evaluated in the context of fairness. This involves fairness to high priority real time flows and fairness to low priority non-real time flows. Additionally, fairness in the short term and long term will be considered.

More investigation also needs to be done regarding the nrt reduced service. The particular model used for service degradation could be further refined to target particular flows. This also might include other QoS parameters so that the degradation would go beyond simply adding increased packet delay.

We also plan to perform a more in depth analysis on the dynamic window approach to estimating the current channel bandwidth. In particular, determining if it tracks the change in bandwidth in the most efficient manner. The value to increase the window size (chosen as 1) might also produce better results using a larger constant.

## 8 Conclusion

The ability of a base station to provide Quality of Service support in a wireless network is dependent both on the call admission policy and scheduler implementation. The CAC algorithm should admit calls to best utilize the available bandwidth. This involves the ability to support the new call both in terms of buffer space, and in terms of traffic requirements such as bandwidth, delay, and packet loss.

A call admission algorithm that takes advantage of the variations in traffic classes has been developed in this paper. This policy, *Threshold Access Sharing*, was chosen from the simulation and comparison of a number of different admission policies. The algorithm operates by admitting traffic according to three different levels. Under low channel utilization all calls are admitted equally. When channel utilization approaches capacity, the lower priority calls are admitted conditionally, knowing they may receive reduced service at a later time. Finally, once the channel has been fully utilized, over-allocation is permitted for high priority traffic by reducing the service level for lower priority traffic.

This call admission policy, used in conjunction with the MPFQ base station scheduling algorithm, can provide per-flow bounded guarantees on the QoS parameters. Thus the wireless channel can be effectively used as a last hop extension to the wired network with QoS guarantees.

## References


[1] S. Sen, A. Arunachalam, K. Basu, and M. Wernik, "A QoS management framework for 3G wireless networks," in *WCNC '99*, (New Orleans, LA), Sept. 1999.

[2] J. Moorman and J. Lockwood, "Multiclass priority fair queuing for hybrid wired/wireless quality of service support," in *WOWMOM '99*, (Seattle,Washington), pp. 43–50, Aug. 1999.

[3] "Traffic Management Specification v4.0." ATM Forum Document AF-TM-0056.000, Apr. 1996.

[4] C.-J. Ho and C.-T. Lea, "Improving call admission policies in wireless networks," *Wireless Networks*, vol. 5, pp. 257–265, Aug. 1999.

[5] T. Kwon, Y. Choi, C. Bisdikian, and M. Nagshineh, "Call admission control for adaptive multimedia in wireless/mobile networks," in *WOWMOM '98*, (Dallas,Texas), pp. 111–116, Aug. 1998.

[6] N. K. Kakani, S. K. Das, S. K. Sen, and M. Kaippallimalil, "A framework for call admission control in next generation wireless networks," in *WOWMOM '98*, (Dallas,Texas), pp. 101–110, Aug. 1998.

[7] B. Li, C. Lin, and S. T. Chanson, "Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks," *Wireless Networks*, vol. 4, pp. 279–290, July 1998.

[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*, vol. 1, pp. 147–160, Feb. 1995.

[9] Z. Ali, E. K. P. Chong, and A. Ghafoor, "A scalable call admission control algorithm for ATM networks," in *GLOBECOM '99*, (Rio de Janeiro, Brazil), pp. 1648–1654, Dec. 1999.

[10] V. K. Garg and J. E. Wilkes, *Wireless and Personal Communications Systems*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[11] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," in *Proceedings of the Winter Simulation Conference*, (Coronado, CA), pp. 597–604, December 1996.

[12] "*Ultra*SAN User's Manual." Center for Reliable and High-Performance Computing, Coordinated Science Laboratory, University of Illinois, 1995.