Efficient Network Utilization for Multimedia Wireless Networks

Xin Liu, Edwin K.P. Chong, and Ness B. Shroff

School of Electrical and Computer Engineering
Purdue University, Lafayette IN 47907, USA
Tel: +1 765 494-1744, Fax: +1 765 494-3358
{xinliu,echong,shroff}@ecn.purdue.edu

Abstract. In this paper, we present an access scheme to satisfy the QoS requirements for two classes of traffic during the *contention phase* of packet-switched wireless communications. In the proposed scheme, different classes of users contend with other users for resources based on controlled class-dependent permission probabilities. We prove that our algorithm is stable for a large class of arrival processes. Under certain QoS requirements, we derive an upper-bound on the throughput for a general class of random access algorithms. We show that the throughput of our algorithm asymptotically approaches this upper-bound. We also show, through numerical examples, that our algorithm achieves high network utilization.

1 Introduction

The goal of wireless communications is to provide a convenient and economical way for people to transfer all kinds of information, such as voice and data. Compared with circuit switching, packet switching provides more efficient multiplexing of different classes of traffic. In circuit switched networks, when a user is admitted to the network, a certain amount of network resource is assigned to the user and exclusively used by the user until its communication finishes, regardless of whether the user has information to transmit during this period. In packet switched networks, when a new user is admitted, no specific resource is assigned to it. Resources are shared by users in the system. A user only occupies the network resource when it has information to transmit. Consider a phone call as an example. When the user talks, voice packets are generated at a certain rate; when the user is silent, no voice packet is generated. On average, the user talks less than half of the entire call duration. In circuit switched networks, the networks assign the voice user the resource equivalent to its packet rate during talking, hence about half of the resources are wasted. In packet switched networks, when a user does not talk, no resource is assigned to this user; when the user begins talking after a period of silence, the network assigns resource to this user again. Hence, in general, packet switching utilizes network resources more efficiently than circuit switching. Efficiency is very important for wireless networks because wireless bandwidth is scarce. However, wireless packet switching

C.G. Omidyar (Ed.): MWCN 2000, LNCS 1818, pp. 108–122, 2000.

suffers from access problems in the uplink. In other words, when a user becomes active, it has packets to transmit and no network resource is assigned to it, the user has to compete with other users to gain the access to network resources. To solve this problem, a variety of contention and reservation medium access control (MAC) protocols have been widely used in the area of communication networks [2, 3, 4, 5]. Typically, there are two transmission phases:

1. Newly activated users compete to gain access to the networks. The first packet of a newly activated user is transmitted through the network using some random access protocols; i.e., contention-based communications. This first packet may be a packet in a special form or a normal data packet. In this paper, we call the first packet a *request*. If the first packet is lost during transmission, or is received in error, then it is retransmitted until successful.
2. Following the first successful contention-based transmission, subsequent transmissions are scheduled contention-free using a scheduling strategy.

We call the first phase the *contention phase* and the second phase the scheduling phase. In this paper, we focus on the contention phase of communications. In packet switched wireless networks, the contention phase may exist throughout the whole communication period, and not only during the admission period. Every time a user becomes active (say, a user begins talking after being silent), at that very moment, because no resource is assigned to the user, the user has to inform the base station about its resource requirement through contention-based communication. Hence, contention-based communication plays an important role in packet-switched wireless networks.

In packet switched networks, admission control and resource allocation are used to provide QoS. In general, admission control is based on the resource allocation scheme. In wired networks, resource allocation is implemented by smart scheduling schemes. However, smart scheduling is not enough to provide QoS for wireless networks, where contention plays an important part. For example, we want to provide delay guarantee to real-time traffic in wireless networks. When a user begins talking, it first sends its request to the base station through random access; i.e., contention-based transmission. Then the base station schedules the traffic after it receives a resource request from the user. Therefore, the user experiences delay caused by contention plus the delay caused by scheduling. To guarantee the delay experienced by the user, we need to guarantee the delay in both contention phase and scheduling phase. During the scheduling phase (if one actually exists in the given implementation), smart scheduling strategies can be used to provide delay guarantees. However, we also need algorithms in the contention phase to provide delay guarantees to users. To provide QoS in the contention phase is intrinsically difficult due to the nature of random access. While there is a significant body of work on the development of effective scheduling and admission control policies to ensure QoS, there is very little work done in implementing QoS during the contention phase of communication.

In this paper, we present an algorithm that implements QoS requirements for two classes of traffic in the contention phase of packet switched time-slotted wireless networks. Controlled time-slotted ALOHA is the random access algorithm

considered in this paper. Two traffic classes, voice and data, are considered. We consider only two classes for simplicity of exposition, convenience of calculation and explanations, although more classes can be similarly considered. We assume that voice users have *delay requirements* and data users do not have such requirements.

In wire-line networks, if two or more users transmit at the same time through the same media, usually all of the transmissions are assumed to have failed. However, this assumption may be unnecessarily pessimistic in the mobile radio environment, where the received packets at the base station are subject to the near/far effect and channel fading. Packets from different users in the same slot may arrive at the base station with different power levels and the base station may successfully decode one or more packet. This is referred to as *capture*. Due to the page-limit requirement of this conference, we only present the QoS algorithm for systems that do not exploit capture. However, the proposed algorithm works for systems with capture too [6]. It is obvious that the system throughput will be improved if the system explores capture. However, unfairness exists between near and far users due to the nature of radio transmission. To achieve fairness and good throughput, we present a distance-dependent permission probability scheme that require users at different distances from the base station to transmit with different probabilities to provide certain delay guarantees, *distance fairness*, and good throughput. In summary, if we do not consider the ability of capture, the QoS requirement is presented in terms of delay. When we consider capture, the QoS requirement is explained in terms of delay and distance fairness.

This paper is organized as follows. In Section 2, we describe the system model. We present and analyze the QoS algorithm in Section 3. An upper-bound for the throughput is derived, under certain QoS requirements, for a general class of random access algorithms. The throughput of our algorithm asymptotically approaches this upper-bound. Simulation results are provided in Section 4. Conclusion and future work are presented in Section 5.

2 System Model

In this section we describe the system model. There is a base station with mobile users in its coverage area. We consider the uplink of a time-slotted system and focus on the contention phase of communication. We assume that time is divided into frames and each frame consists of M request slots. Each request slot is large enough to contain a fixed size request. The base station monitors and controls the contention phase in the system. *In the following, when we mention users we mean newly activated users with requests to transmit, except otherwise specified.*

At the beginning of a frame, the base station broadcasts a permission probability for each class of users through a non-collision error-free signaling channel. A user decides whether or not to transmit in a request slot in the frame according to the permission probability of its class broadcasted by the base station. Different classes of users may have different permission probabilities.

We assume that a user can transmit at most once in a frame. There are M request slots in each frame. The parameter, M, determines how often the base station updates its control parameters, and how long a user waits before it retransmits. In practice, the larger the value of M, the less the signaling, the better the estimation of the number of users, however, the longer the delay.

In some cases, we prefer a large value of M. An example of such a scenario is in satellite communications. After the contention of a time slot, a user cannot know immediately whether its request is successfully received by the hub station. In satellite communications, the round trip delay is relatively large. For instance, the propagation delay is around 20–25ms for LEO (low earth orbit) systems [8]. An immediate ack from the the hub is impossible. Furthermore, the coverage area of satellite communications is relatively large, it is difficult for an earth station to detect whether its transmission is successful. Hence, a large value of M may be suitable for such a case. In other cases, a small value of M could be favored. A good example of such a case is a local wireless network, where the sum of the round trip delay, and processing time, etc., is small. A user transmits, then waits for acknowledgment. If the user does not receive an acknowledgment from the base station in the predetermined waiting time, it assumes that the transmission has failed. The user could retransmit it in the next frame. The extreme case is when $M = 1$; i.e., a user can retransmit its request in the next request slot. In the extreme case $M = 1$, the scheme studied in this paper becomes the pure priority scheme; i.e., when there are voice users, no data user transmits, and when there is no voice user, data users transmit. However, even in a wireless LAN, it is not necessary to adopt such a small value of M. Usually, the requests are much shorter than normal data packets. Hence, the delay caused by several request slots are tolerable in order to reduce the cost of extensive signaling.

In this paper, we assume that the system is not capable of correctly deciphering any transmissions when two or more overlapping transmissions arrive in the same slot; i.e., if two or more users transmit their requests through the same request slot in a frame, neither of them can be successfully received. This situation is called collision.

We assume that a request is never discarded; i.e., a user always retransmits its request until it is acknowledged by the base station that its request has been received successfully. While the request of a user is delayed, some packets may be buffered at the user. In real-time applications, human factors may decide whether to send a delayed packet or to drop it. This issue is irrelevant in our scheme. Furthermore, we assume that the acknowledgment is error-free and the base station uses a scheduling strategy to decide when the active user should transmit in the reservation phase of communication.

3 The QoS Algorithm

We first present the QoS algorithm with restriction to the delay requirement of voice users. We, then, analyze the throughput and stable condition. Finally, we

derive a throughput upper-bound under the QoS requirement for a large class of random access algorithms.

3.1 Algorithm

Let p_v (p_d) denote the permission probability that a voice (data) user transmits in a request slot in a frame. In this paper, the permission probabilities, p_v and p_d, are used to stabilize the ALOHA system, to achieve good throughput, and to provide QoS guarantees. The use of permission probabilities to stabilize ALOHA is not a new idea. Permission probabilities are also used to provide priority to voice users in [4, 7]. In the literature, there are algorithms, centralized and decentralized, to estimate the number of users in the system. All these algorithms can be used in our scheme. Hence, we focus on how to use the permission probabilities to satisfy QoS instead of how to estimate the number of users. During the analysis we assume that the base station knows the precise numbers of voice users and data users in each frame. Knowing this information is the ideal condition of the algorithm. Practically, we use a Kalman filter to estimate the numbers of voice users and data users with requests in each frame. We show through simulations that using a Kalman filter for the estimation provides very good results.

As mentioned before, a user can transmit at most once in a frame. We do not distinguish between newly arrived and retransmitted users. The base station broadcasts p_v and p_d at the beginning of frame i. A voice user randomly selects a request slot to transmit in this frame with probability p_v, as would a data user with probability p_d. All users select and transmit independently. The base station acknowledges those users whose requests have been successfully accepted at the end of frame i. Users that have not been acknowledged assume that their requests have not been successfully transmitted. They retransmit in the next frame. The base station estimates the number of users in the system, calculates p_v and p_d for frame $i+1$, and so on. It is easy to prove that the throughput is maximized when M users transmit in each frame [6]. However, this throughput may come at the cost of excessive delay for voice users. Hence, we need to develop a scheme that attempts to maximize throughput subject to a given level of delay requirement for voice users.

A good measure of QoS is the delay experienced by a user before its request is successfully received by the base station. However, the precise delay distribution of voice users is very difficult to find in this context. Thus, we define an average success probability, $\bar{P}_s$, as the QoS measure used in this paper. Suppose the system has reached steady state. When a voice user becomes active, on average, it transmits its request successfully with probability $\bar{P}_s$, given by

$$\bar{P}_s := E\left[p_s(N_v, N_d)\right] = \sum_{i,j} p_s(i,j)\pi(i,j), \tag{1}$$

where $p_s(i,j)$ is the probability that a voice user transmits its request successfully in a frame in steady state when there are i voice users and j data users in the

system, and $\pi(i, j)$ is the steady state distribution that i voice users and j data users are in the system.

Our QoS requirement for voice users is $\bar{P}_s \geq A_0$, where A_0 is the given delay threshold. Roughly speaking, the contention delay of a voice user is geometrically distributed with parameter $\bar{P}_s$; i.e., the distribution of access delay D is approximated by $P(D = x) = \bar{P}_s(1 - \bar{P}_s)^{x-1}$. When the correlation of the numbers of users between cells is small, the approximation is good. If the number of users arrived in each frame is independent, then the larger the M, the better the approximation. In Section 4, we show that the distribution of voice users from simulations is well approximated by a geometric distribution (see Figure 1).

The QoS algorithm is described as follows. Suppose that the base station knows that N_v voice users and N_d data users are in the system. Then, the permission probabilities of voice users and data users are

$$p_v = \min(1, \frac{M}{N_v}),$$
$$p_d = \begin{cases} \min\left(1, \frac{(C-N_v)^+}{N_d}\right) & : \text{ if } N_v > 0, \\ \min\left(1, \frac{M}{N_d}\right) & : \text{ if } N_v = 0, \end{cases} \tag{2}$$

where

$$(x)^+ = \begin{cases} x & : \text{ if } x \geq 0, \\ 0 & : \text{ otherwise.} \end{cases}$$

Note that C is a tuning parameter used to satisfy the QoS requirements of voice users. So the algorithm does the following. If the number of voice users in the system is less than M, all voice users can transmit freely. In this case, data users may or may not be allowed to transmit. If the number of voice users in the system is greater than M, then a voice user is allowed to transmit based on the outcome of the toss of a biased coin with probability M/N_v of success. In this case, no data users are allowed to transmit. Before we illustrate how to calculate C, we first make a few observations:

- Data users yield to voice users the right to access request slots.
- The parameter C satisfies $0 \leq C \leq M$. The expected number of data users to transmit is $(C - N_v)^+$. The total throughput is maximized when $C = M$. The larger the value of C, the higher the throughput, and the larger the delay of voice users. Hence, there is a tradeoff between the throughput of the system and the delay requirement of voice users. When the QoS requirement is stringent, C is small, data users are allowed to access request slots with lower probability, and voice users have a higher probability to succeed in a frame.
- When there is no voice user; i.e., $N_v = 0$, the value of p_d is set to maximize the throughput.

The tuning parameter C can be calculated theoretically. A two dimensional Markov chain is used to calculate the steady-state distribution. Suppose that we

know the distribution of the arrival process. Let $C = x, 0 \leq x \leq M$. Transmission probabilities between states are determined by (2) and the arrival process. Hence, $\pi(i,k)$ can be calculated and so can $\bar{P}_s$. Since $\bar{P}_s$ is a monotone decreasing function of x, denoted as $\bar{P}_s(x)$ for $0 \leq x \leq M$, the parameter C is the unique root of $\bar{P}_s(x) = A_0$, which can be obtained easily using standard zero-finding algorithms. If $\bar{P}_s(0) < A_0$, the QoS requirement cannot be satisfied. In other words, even without data users, the delay caused by the contention among voice users are still larger than required when $\bar{P}_s(0) < A_0$.

Practically, there is a very simple approximation for C. Let K_0 satisfy

$$\left(1 - \frac{1}{M}\right)^{K_0 - 1} = A_0. \tag{3}$$

If K_0 is not too small compared to M and the fraction of voice users is not too large, then K_0 is a good approximation of C. In this case, the number of voice users in the system in steady state is seldom larger than K_0. Therefore, the average delay $\bar{P}_s$ is:

$$\begin{aligned}\bar{P}_s &= E(p_s) = E(p_s(i)|i \leq C)p(i \leq C) + E(p_s(i)|i > C)p(i > C) \\ &\approx (1 - \frac{1}{M})^{C-1} = (1 - \frac{1}{M})^{K_0 - 1} = A_0.\end{aligned}$$

In fact, if $K_0 \geq 0.5M$ and the fraction of voice users is less than 70%, $C \approx K_0$ is a good approximation. We set $C = K_0$ in simulations in Section 4 and find that it works well.

We, next, analyze the algorithm. First, we calculate the throughput. Second, we prove that the algorithm is stable for a large class of arrival processes. Then, we derive an upper bound on the throughput of random access algorithms under the QoS requirement $\bar{P}_s \geq A_0$. We show that the throughput of our algorithm asymptotically approaches the upper-bound.

3.2 Throughput

Suppose that there are k users transmitting in a frame. Each user selects one of the request slots randomly and independently. When only one user transmits in a request slot, we assume that the transmission is successful. When two or more users transmit in the same request slot, we assume that neither of the transmission is successful. The throughput, T_k, is defined as the average number of requests that are successfully transmitted in a frame and p_k is the probability that a user transmits successfully. We then have

$$\begin{aligned}T_k &= k\left(1 - \frac{1}{M}\right)^{k-1}, \\ p_k &= \frac{T_k}{k} = \left(1 - \frac{1}{M}\right)^{k-1}.\end{aligned}$$

We consider the throughput under three conditions:

1. When $N_v \geq C$, each voice user transmits in a request slot with probability $p_v = \min(1, M/N_v)$ and no data user transmits. The throughput is:

$$\begin{aligned} T(N_v, N_d) &= \sum_{i=0}^{N_v} T_i P(i \text{ voice users transmit in this frame}) \\ &= N_v p_v \left(1 - \frac{p_v}{M}\right)^{N_v - 1}. \end{aligned} \tag{4}$$

2. When $N_v < C$, each voice user transmits in a request slot with probability 1 and each data users transmits with probability $p_d = (C - N_v)/N_d$. Therefore,

$$\begin{aligned} p_s(N_v, N_d) &= \sum_{i=0}^{N_d} p_{i+N_v} P(i \text{ data users transmit in this frame}) \\ &= \left(1 - \frac{1}{M}\right)^{N_v - 1} \left(1 - \frac{p_d}{M}\right)^{N_d}. \end{aligned}$$

The throughput consists of successfully transmitted voice and data requests:

$$\begin{aligned} T(N_v, N_d) &= \sum_{i=0}^{N_d} T_{i+N_v} P(i \text{ data users transmit in this frame}) \\ &= N_v \left(1 - \frac{1}{M}\right)^{N_v - 1} \left(1 - \frac{p_d}{M}\right)^{N_d} \\ &\quad + (C - N_v) \left(1 - \frac{1}{M}\right)^{N_v} \left(1 - \frac{p_d}{M}\right)^{N_d - 1}. \end{aligned} \tag{5}$$

3. When $N_v = 0$, data users transmit with probability p_d, $p_d = min(1, M/N_d)$, to maximize the throughput.

$$T(0, N_d) = N_d p_d \left(1 - \frac{p_d}{M}\right)^{N_d - 1}. \tag{6}$$

3.3 Stability Analysis

We now prove that our algorithm is stable with a fairly weak assumption on the arrival process. We consider a system with a unique stationary distribution as a stable system. We use Pake's Lemma to find a sufficient condition for the system to be stable [9].

Lemma 1 (Pake's Lemma). *Let $\{X_k, k = 0, 1, 2, \cdots\}$ be an irreducible, aperiodic homogeneous Markov chain with state space $\{0, 1, 2, \cdots\}$. The following two conditions are sufficient for the Markov chain to be ergodic.*

a) $|E(X_{k+1} - X_k | X_k = i)| < \infty, \ \forall\ i,$

b) $\limsup_{i \to \infty} E(X_{k+1} - X_k | X_k = i) < 0.$

Note that an irreducible, aperiodic, ergodic Markov chain has a unique stationary distribution.

Let A_k be the total number of users that arrive in the kth frame. Suppose that $\{A_k, k = 0, 1, 2, \cdots\}$ are random variables with mean value λ. Let X_k be the number of users (voice users and data users) at the beginning of the kth frame, then $X_k = N_v + N_d$. Let $B(X_k)$ be the number of users whose requests are successfully transmitted in the kth frame. We now prove that $\{X_k, k = 0, 1, 2, \cdots\}$ is ergodic using Pake's lemma. We have

$$X_{k+1} = X_k + A_k - B(X_k).$$

So, for any i,

$$\begin{aligned} &|E(X_{k+1} - X_k | X_k = i)| = |E(A_k - B(X_k) | X_k = i)| \\ &= |E(A) - E[B(i)]| \leq |E(A)| + |E[B(i)]| \leq \lambda + M < \infty. \end{aligned}$$

Hence, condition (a) of Pake's lemma is satisfied.

To satisfy condition (b) of Pake's lemma, we require that

$$\begin{aligned} &\limsup_{i\to\infty} E(X_{k+1} - X_k | X_k = i) \\ &= \limsup_{i\to\infty} E(A_k - B(X_k) | X_k = i) \\ &= \limsup_{i\to\infty} (\lambda - E[B(i)]) < 0. \end{aligned}$$

So

$$\lambda \leq \liminf_{i\to\infty} E[B(i)] \tag{7}$$

is a sufficient condition for the system to be stable.

In our QoS algorithm, when there are N_v voice users and N_d data users, the total number of users is $i = N_v + N_d$. Then, there exists an L such that for all $i \geq L$, we have [6]:

$$T(N_v, N_d) \geq C \left(1 - \frac{C}{iM}\right)^{i-1}.$$

Hence,

$$B(i) \geq C \left(1 - \frac{C}{iM}\right)^{i-1}.$$

Then

$$\liminf_{i\to\infty} E[B(i)] \geq \liminf_{i\to\infty} C \left(1 - \frac{C}{iM}\right)^{i-1} = Ce^{-\frac{C}{M}}. \tag{8}$$

Hence, from (8), $\lambda \leq Ce^{-C/M}$ is the sufficient condition for the system to be stable under the QoS requirement $\bar{P}_s \geq A_0$, where λ is the arrival rate. Note that

in the special case $C = M$; i.e., the system is designed to achieve the maximum achievable throughput, (8) becomes:

$$\liminf_{i\to\infty} E[B(i)] = \liminf_{i\to\infty} M\left(1-\frac{1}{i}\right)^{i-1} = Me^{-1} \tag{9}$$

The sufficient stable condition is $\lambda < Me^{-1}$, which is exactly the stable condition for slotted ALOHA. Furthermore, there is no bistable point in the system because the throughput does not decrease when the number of blocked users in the system increases.

3.4 Upper Bound on Throughput

We consider the QoS requirement as $\bar{P}_s \geq A_0$. With this restriction, we derive an upper-bound on the throughput for random access algorithms satisfying the following two assumptions. First, all users transmit in request slots randomly and independently. Second, each user transmits in at most one request slot in each frame. Let Ω be the set of all such random access algorithms.

We consider the throughput under two conditions. Condition 1: there is at least one voice user in the system. Condition 2: there is no voice user in the system. First, we consider the throughput under Condition 1. Let X denote the total number of users that transmit in this frame, $1 \leq X \leq \infty$. The probability that the voice user successfully transmits its request in this frame is p.

$$p = \begin{cases} p_X & : \text{ if the user transmits in this frame,} \\ 0 & : \text{ otherwise,} \end{cases}$$

where

$$p_X := \left(1-\frac{1}{M}\right)^{(X-1)}.$$

Note that

$$E(p_X) = E\left(\left(1-\frac{1}{M}\right)^{(X-1)}\right) \geq E(p) = \bar{P}_s \geq A_0. \tag{10}$$

Let T_1 be the throughput given that there is at least one voice user in the system. Then,

$$T_1 = E\left(X\left(1-\frac{1}{M}\right)^{(X-1)}\right). \tag{11}$$

We want to maximize (11) with the constraint (10). Let $Y = (1-1/M)^{(X-1)}$. So

$$E(Y) \geq \bar{P}_s = A_0 = \left(1-\frac{1}{M}\right)^{K_0-1}.$$

Let $f(y) = -y\left(\frac{\ln y}{\ln\left(1-\frac{1}{M}\right)} + 1\right)$, which is a strictly convex function. By Jensen's inequality [1],

$$T_1 = E\left(-f(Y)\right) \leq -f(E(Y)) = K_0\left(1 - \frac{1}{M}\right)^{K_0-1} =: T_C. \tag{12}$$

Next, we consider the condition 2; i.e., no voice user is in the system. Let T_0^α be the throughput of a random access algorithm α when there is no voice user in the system. Let $T_0^m = \max\{T_0^\alpha,\ \alpha \in \Omega\}$. Let q_α denote the probability that no voice user is in the system of a random access algorithm α. Let $P_0 = \max\{q_\alpha,\ \alpha \in \Omega\}$. For algorithm α, let P_1^α be the probability that there is at least one voice user in the system. Hence, $1 - P_1^\alpha \leq P_0$. The throughput T of algorithm α is given by:

$$\begin{aligned} T &= T_1 P_1^\alpha + T_0^\alpha(1 - P_1^\alpha) \leq T_C P_1^\alpha + T_0^m(1 - P_1^\alpha) \\ &= T_C + (1 - P_1^\alpha)(T_0^m - T_C) \leq T_C + P_0(T_0^m - T_C) =: T_{max}. \end{aligned} \tag{13}$$

Therefore, T_{max} is the upper-bound on the throughput of random access algorithms in Ω (algorithms such that all users transmit for request slots randomly and independently, and each user transmits for at most one time slot in a frame). This upper-bound is not restricted to the (p_v, p_d) strategy used in this paper.

The above upper-bound, T_{max}, may not be tight. We compare T_1 with T_C. Since f is a strictly convex function, (12) achieves equality when $Y = E(Y)$ with probability 1. Hence, the upper-bound T_C is only achievable if $X = C$ with probability 1; i.e., there are always exactly C users transmitting in each frame. However, C may not be an integer and it may not be possible to let exactly C users transmit in random access algorithms. So T_{max} may not a tight upper-bound. We try to approach the upper-bound by assigning p_v and p_d such that $E(N_v p_v + N_d p_d) = C$ in our scheme.

Next, we show that

$$\lim_{M\to\infty} \frac{T(N_v, N_d)}{T_{max}} = 1,$$

when there are enough users in the system.

With some tedious algebra [6], we can show that

$$T(N_v, N_d) \geq T_C\left(1 - \frac{1}{M}\right),$$

when

$$N_v + N_d \geq C.$$

Recall that P_0 is the maximum probability that there is no data user in the system. Let p_0 be the probability that there is no new voice user with a request in a frame. Then, $P_0 = p_0 P(\text{all voice users with requests transmit successfully by the end of a frame in steady state})$. In practice, P_0 is small when M is large; i.e.,

in a large frame, it is unlikely there is no voice user in the frame. For example, if the arrival process of voice users is a Poisson process with mean vM, then $P_0 \leq p_0 = e^{-vM}$. Suppose $P_0 \to 0$ as $M \to \infty$. We have

$$\frac{T(N_v, N_d)}{T_{max}} \geq \frac{T_C\left(1 - \frac{1}{M}\right)}{T_C + P_0(T_0^m - T_C)} \to 1.$$

Hence, the throughput of the presented QoS algorithm asymptotically approaches the upper-bound. In other words, when there is at least one voice user in the system, the throughput of our QoS algorithm approaches T_C. Furthermore, as M goes large, the probability that there is no voice user in the system goes to zero. So the throughput of our QoS algorithm asymptotically approaches the upper-bound.

4 Simulation Results

In this section, we provide simulation results of the proposed scheme. For all simulations in this section, we set $M = 20$; i.e., there are 20 request slots in each frame. For each figure, we run simulation for $100,000$ frames in a single-cell. We assume the arrival processes of voice users and data users are independent Poisson processes with the same average rate.

At the beginning of each frame, the base station announces N_v and N_d, the numbers of voice users and data users in the system. (The announced numbers are estimated by the base station in the practical approach.) Knowing N_v and N_d, each user decides its transmission probability according to (2). With this probability, the user selects and transmits in a request slot in the frame. If the user is the only one transmitting in its request slot, its transmission is successful. Otherwise, the user has to wait for the next frame to retransmit and its delay is increased by one. The unit of delay is frame.

Figure 1 indicates the delay distribution of a voice user when $A_0 = 0.6$, where A_0 is the required success probability. We can see that the delay distribution of a voice user is well approximated by a geometric distribution when the numbers of new arrived users at different frames are independent and $M = 20$. Hence, in other figures, we use the success probability as the delay performance measure.

Figures 2 and 3 illustrate the performance of the proposed QoS algorithm. Figure 2 indicates the delay performance of voice users. The delay performance is shown by the average probability of success. Simulations are run under both the ideal condition and the practical condition. By the ideal condition, we mean that the base station knows the exact numbers of voice and data users in the system. In practice, a Kalman filter is used to estimate the numbers of users. The Kalman filter approach is implemented with two threshold values. We use (3) to approximate C. In the ideal condition, (3) offers a pretty good approximation. With $C = K_0$ in the Kalman filter approach, $\bar{P}_s$ is less than the QoS requirement due to estimation errors. Thus, in practice, we should use a smaller threshold value than the one calculated under the ideal condition, which is represented by the curve with $C = 0.9K_0$. Figure 3 shows the throughput performance. It is

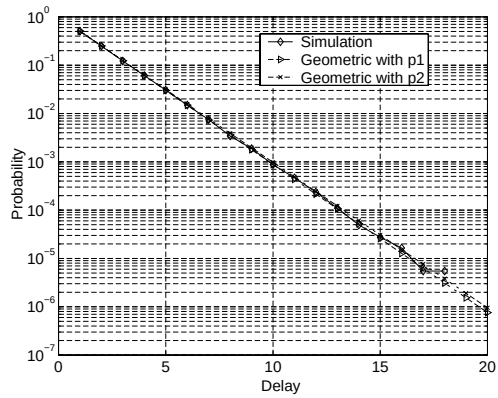

Fig. 1. Delay distribution of a voice user when $M = 20$, $p1$ is the reciprocal of the average delay of voice users, and $p2$ is the average probability of success of voice users.

obvious that the throughput decreases with the increase of A_0. We compare the throughput in the ideal condition with the practical approaches. As expected, the Kalman filter approach with the smaller C has less throughput, illustrating the tradeoff between the throughput and QoS. We use the probability of no new voice user in a frame, $p_0 = e^{-r\lambda}$, as the upper-bound of P_0, where P_0 is the probability of no voice user in a frame. Hence, p_0 is used to calculate the upper-bound of throughput shown in Figure 3, which results a looser upper-bound than that in (13). However, we still note that in most cases, the throughput in the ideal condition is quiet close to the upper-bound in the figure.

5 Conclusions

We present a random access scheme that provides certain QoS guarantees during the contention phase of communication. Permission probabilities are used to provide QoS for two traffic classes, voice users and data users. The same idea can be extended to multi-class users. The QoS requirement of voice users is defined as $\bar{P}_s$, the average success probability of voice users. For a predetermined QoS measure $\bar{P}_s$, a threshold C is calculated such that a voice users has an average success probability larger or equal to $\bar{P}_s$. We prove that the algorithm is stable with a weak assumption. We derive the upper-bound of a general class of random access algorithms under the QoS requirement in term of $\bar{P}_s$ and show that the studied algorithm asymptotically approaches the upper-bound. The analysis is based on the QoS algorithm without capture. Note that the QoS algorithms with and without capture are the same in essence except that the success probability is higher when capture is considered [6].

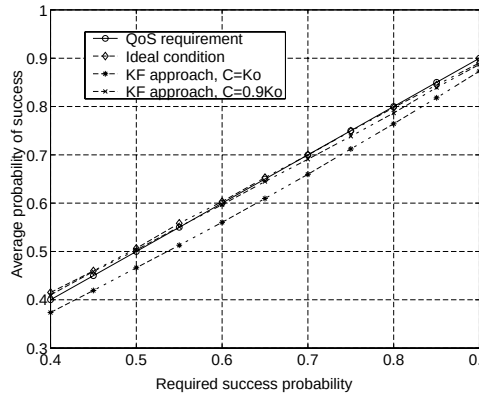

Fig. 2. Delay performance without capture for $M = 20$ with 50% voice users. In the legend, KF denotes Kalman filter.

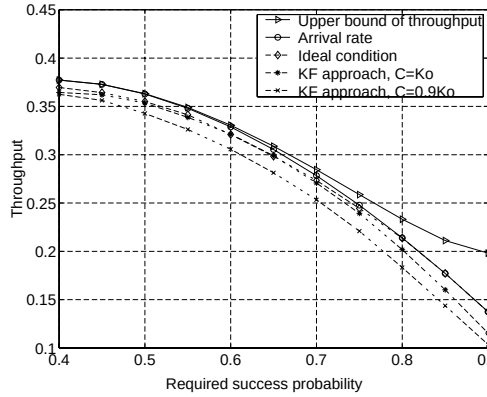

Fig. 3. Throughput without capture for $M = 20$ with 50% voice users. In the legend, KF denotes Kalman filter.

In wireless networks, providing QoS during contention phase is important to support bursty traffic. It is quite different from the wire-line scenario. So existing methods such as using in ATM do not apply directly. There would be large research space for this topic.

References

[1] P. Billingsley, *Probability and Measure*. Wiley, 1985.
[2] C. Bisdikian, "A review of random access algorithms," *IBM Res. Rep.*, no. RC20348, Jan. 1996.

[3] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 885–890, 1989.
[4] W. S. Jeon, D. G. Jeong, and C.-H. Choi, "An integrated service MAC protocol for local wireless communications," *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 352–363, 1998.
[5] R. LaMaire, A. Krishna, and H. Ahmadi, "Analysis of a wireless MAC protocol with client-server traffic and capture," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 8, pp. 1299–1313, 1994.
[6] X. Liu, E. Chong, and N. Shroff, "An access scheme to provide qos in packet-Switched wireless networks," Tech. Rep., Purdue University, 2000.
[7] K. Mori and K. Ogura, "An adaptive permission probability control method for integrated voice/data CDMA packet communications," *IEICE Trans. Fundamentals*, vol. E81-A, no. 7, pp. 1339–1348, 1998.
[8] H. Peyravi, "Medium access control protocols performance in satellite communications," *IEEE Communications Magazine*, vol. 37, no. 3, pp. 62–71, 1999.
[9] R. Rom and M. Sidi, *Multiple access protocols : performance and analysis.* New York : Springer-Verlag, 1990.