Design and performance evaluation of an RRA scheme for voice-data channel access in outdoor microcellular environments

Allan C. Cleary [a] and Michael Paterakis [b]

[a] *CIS Department, University of Delaware, Newark, DE, USA*
[b] *Electronics and Computer Engineering Department, Technical University of Crete, 731-00 Chania, Crete, Greece*

In PCS networks, the multiple access problem is characterized by spatially dispersed mobile source terminals sharing a radio channel connected to a fixed base station. In this paper, we design and evaluate a reservation random access (RRA) scheme that multiplexes voice traffic at the talkspurt level to efficiently integrate voice and data traffic in outdoor microcellular environments. The scheme involves partitioning the time frame into two request intervals (voice and data) and an information interval. Thus, any potential performance degradation caused by voice and data terminals competing for channel access is eliminated. We consider three random access algorithms for the transmission of voice request packets and one for the transmission of data request packets. We formulate an approximate Markov model and present analytical results for the steady state voice packet dropping probability, mean voice access delay and voice throughput. Simulations are used to investigate the steady state voice packet dropping distribution per talkspurt, and to illustrate preliminary voice-data integration considerations.

1. Introduction

Personal communication services (PCS) embody the notion that access to land-based public networks should reside with the individual rather than with a fixed location (e.g., a wall phone jack), thereby permitting any user to initiate or accept calls from anywhere. An overview of PCS and a survey of the current research and development can be found in [1] and [2], respectively. In this paper, we design and evaluate a multiple access scheme that multiplexes voice traffic at the talkspurt (vocal activity) level to efficiently integrate voice and data traffic in outdoor microcellular environments.

In PCS networks, the multiple access problem is characterized by spatially dispersed mobile source terminals sharing a radio channel connected to a fixed base station [3]. A well designed multiple access scheme will reduce system costs by maximizing system capacity; satisfy quality of service requirements such as voice packet dropping probability and access delay; and, integrate different classes of traffic such as voice and data. These straightforward design goals are complicated by the bandwidth limitations of the radio channel and by the contradictory requirements of voice and data traffic. Multiple access schemes proposed for PCS usually involve Frequency-Division Multiple Access (FDMA), Time-Division Multiple Access (TDMA), Code-Division Multiple Access (CDMA) or combinations thereof.

Operating over a time-slotted channel with a periodic frame structure, Reservation Random Access (RRA) protocols combine a random multiple access algorithm (e.g., slotted Aloha) with TDMA [4]. In general, the time slots may be viewed (explicitly or implicitly) as being information or request slots, and the information slots can be classified as being reserved or available. Contending terminals (those with packets and without a reservation) use a random access algorithm to compete for channel resources. After successfully transmitting a request, the terminal receives a reservation for an information slot (or slots). A terminal with a reservation transmits freely during its reserved slot(s) and the reservation is held for as long as it continues to transmit packets in successive frames.

Because RRA protocols provide a way to exploit the on/off characteristics of packetized speech traffic, various forms have been proposed and studied for use in future wireless networks [6–14]. The following discussion is intended to highlight those that have influenced this work, rather than to provide an exhaustive survey of the literature.

In the RRA schemes considered in [5–10], every slot within the frame is an information slot and the contending terminals attempt to transmit their voice (or data) packet into the available information slots. A voice terminal that successfully transmits its packet during an available slot receives a reservation for the corresponding slot in successive frames, until it exits talkspurt. In PRMA [5–7] the contending voice and data terminals both use a generalized slotted Aloha algorithm to access the channel. To ensure that voice terminals have greater access to the available slots, the retransmission probabilities are weighted to favor voice terminals. To increase data throughput, Integrated Packet Reservation Multiple Access (IPRMA) [8] extends the PRMA voice reservation mechanism to data; and, a priority mechanism is provided to ensure that voice packets have greater access to the available slots.

RRA voice protocols that can be used to eliminate collisions between voice and data packet transmissions in the available slots are a promising alternative to PRMA [9,10]. This is accomplished by combining a random access algorithm that identifies the end of the voice contention with a policy to resolve the voice traffic first. Thus, every terminal within the microcell can differentiate between available voice and available data slots and the voice and data random access transmissions can be separated.

In a modification to PRMA, known as PRMA++ [11,12], all of the slots within a frame are of equal size, however a fixed number of slots are designated as request slots. Contending voice terminals follow a generalized slotted Aloha algorithm to transmit their reservation request packets into the request slots. On successful receipt of a request packet, the base station either provides a reservation for an information slot (if available) or it queues the request. In the latter case, the terminal monitors the base-to-mobile channel until it is granted a reservation.

The RRA schemes proposed in [13,14] involve partitioning a portion of the frame into mini-slots. The contending terminals use slotted Aloha to transmit reservation request packets into the mini-slots. The base station provides acknowledgments and allocates channel resources. Voice traffic performance is analyzed in [13] and a variant of dynamic TDMA is investigated in [14]. In [14], voice-data integration is achieved by partitioning the information slots into two intervals; one designated for voice and the other for data traffic.

An inherent inefficiency common to the protocols in [5–12] is that an entire time slot is wasted when a collision is caused by terminals simultaneously contending for channel access. The amount of degradation depends on the packet size (time slot duration) and it increases with the traffic load. The inefficiency shared by the protocols in [10–14] is the fixed overhead due to the incorporation of request slots (i.e., control signaling rather than information traffic). However, a persuasive argument in favor of protocols like [10–14] is that the base station controls the allocation of the channel resources. This centralized control can be exploited to implement access control policies, dynamic channel assignment and/or the integration of different priority traffic classes.

The remainder of this paper is organized as follows. We propose an RRA scheme along the general lines of [13,14] and we present several random access algorithms for transmitting request packets in section 2. In section 3, we list the primary model assumptions; formulate a Markovian model for the evaluation of the voice traffic performance; and derive expressions for the steady state voice packet dropping probability, access delay and throughput. The system parameters used in the analysis and simulations are included in section 4, along with the method for calculating the steady state distribution and a brief discussion of the simulations. We present and discuss representative analytic and simulation results in section 5. The simulation results are used to verify the analysis, to further investigate voice traffic performance, and to illustrate preliminary voice-data integration considerations. The paper is concluded in section 6.

2. RRA scheme

Within the microcell, spatially dispersed source terminals share a radio channel that connects them to a fixed base station. The base station allocates channel resources, delivers feedback information and serves as an interface to the mobile switching center. The mobile switching center provides access to the fixed network infrastructure. Because communication between the base station and the mobile terminals involves one to many transmissions, it is usually conducted over a contention-free TDM broadcast channel [3]. Thus, we focus on the mobile-to-base (many to one) channel.

We assume that voice and data traffic is generated by mobile pedestrians who access the network with small, lightweight and low-power devices (i.e., in the category of low tier PCS [1]). Each voice terminal is equipped with a voice activity detector (VAD) that generates packets during periods of vocal activity (*talkspurt*) and suppresses periods of silence. Thus, since a voice terminal only requires channel access during talkspurt, the time periods corresponding to silence gaps within a conversation can be used to transmit packets from other source terminals (i.e., multiplexing occurs at the talkspurt level).

Voice packet delay requirements are generally more strict than those for data packets, because delays in voice communication are annoying to a listener. Thus, each voice packet must be delivered within a specified maximum delay, $D_{\max}$. Whenever the delay experienced by a voice packet exceeds $D_{\max}$, the voice packet is dropped. Speech can withstand a small (1–2 percent) amount of dropped packets without suffering large quality degradation [15], at least one which can be perceived by humans. On the other hand, data applications are more tolerant of delays (delays up to approximately 200 ms are usually acceptable), but often require 100 percent delivery of correct packets (e.g., a file transfer).

2.1. RRA protocol

The mobile-to-base channel is organized into periodic time frames of fixed duration. The frame duration is selected such that a voice terminal in talkspurt generates exactly one packet per frame. As illustrated in figure 1, each frame consists of one voice request interval, one data request interval and an information interval. Within the information interval, each slot accommodates exactly one, fixed

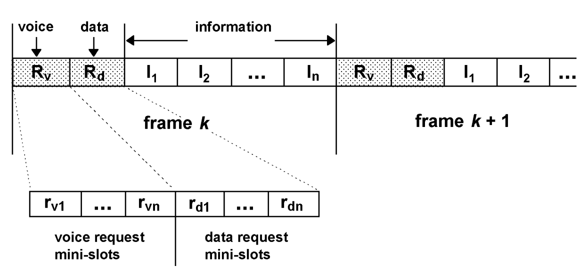


Figure 1. The voice, data and information intervals within a frame.

length, packet that contains voice (or data) information and a header.

Both of the request intervals are subdivided into mini-slots and each mini-slot accommodates exactly one, fixed length, request packet. For both voice and data traffic, the request must include a source identifier. For data traffic, the request might also include message length and quality of service parameters such as priority and required slots/frame. To simplify our presentation, we will assume that both of the request intervals contain an equal number of mini-slots and that the data terminals are given at most one information slot per frame.

Voice (data) terminals with packets, and no reservation, contend for channel resources using a random access protocol to transmit their request packets only during the voice (data) request interval. The base station broadcasts a short binary feedback packet at the end of each mini-slot that indicates only the presence or absence of a collision within the mini-slot (collision (C) versus non-collision (NC)). Since the feedback packet is small (several bits) and the transmission delay within a microcell is negligible, we assume that the feedback information is immediately available to the terminals (i.e., before the next mini-slot). Upon successfully transmitting a request packet the terminal waits until the end of the frame to learn of its reservation slot. If unsuccessful within the request interval, the terminal attempts again in the request interval of the next frame. A terminal with a reservation transmits freely during its reserved slot. Generally, a terminal that fails to transmit a request tries again in successive frames until it succeeds. However, since voice packets that age beyond $D_{\max}$ are dropped, a voice terminal may stop transmitting requests without ever succeeding, because it has transitioned to silence and all of its packets have timed out.

To allocate channel resources, the base station maintains a dynamic table of the active terminals within the microcell. For example, information within the table might include the terminal identifier, the virtual circuit identifier, the channel resources allocated, and quality of service parameters. Upon successful receipt of a request packet, the base station provides an acknowledgment and queues the request. The base station allocates channel resources at the end of the frame, if available. If the resources needed to satisfy the request are unavailable, the request remains queued. Voice terminals with queued requests and data terminals with packets must continuously monitor the base-to-mobile channel. Upon call completion, or when an active terminal exits the microcell (handover), the base station will delete the table entry after some prescribed period of time.

As our focus is on channel access by a fixed number of terminals within the system (steady state), we do not address call set-up and tear down issues. For concreteness, we assume that the base station always allocates the earliest empty information slot within the frame. Next we assume that voice is of higher priority than data traffic. Thus, the base station services every outstanding voice request before servicing any data requests. Within each priority class, the queuing discipline is assumed to be First Come First Served. We note that, depending upon the QOS criteria, other queuing disciplines may be more suitable for the data traffic. Finally, the base station preempts data reservations to service voice requests. Thus, whenever new voice requests are received and every slot within the frame is reserved the base station attempts to service the voice requests by canceling the appropriate number of reservations belonging to data terminals (if any). When data reservations are canceled, the base station notifies the affected data terminal and places an appropriate request at the front of the data request queue.

2.2. Random access algorithms for voice terminals

We present three random access algorithms for the transmission of voice requests and one for the transmission of data requests. Our selection of algorithms stems from the desire to investigate performance aspects of the proposed RRA scheme, rather than optimization for a specific set of system parameters.

2.2.1. Ideal

Every request packet present at the start of the reservation request interval is correctly received by the base station within the duration of the request interval. This protocol is included because it provides an upper bound for the voice system capacity (the maximum number of voice terminals, constrained by less than about 1 % voice packet dropping probability), and a lower bound for the voice access delay (the time between the start of a talkspurt and the end of the first voice packet transmission into a reserved slot).

2.2.2. Slotted Aloha [4]

A contending terminal transmits its request packet only if it has permission to transmit. Permission is issued by a pseudo random number generator with probability, p, in each request slot. We set the system design parameter, p, equal to the constant value of $1/3$.

The value of $1/3$ was chosen to ensure stable operation of the generalized Aloha algorithm. Our voice traffic simulations with $p = 0.5$ indicate that with more than about 75 active voice terminals the system tends to oscillate between two equilibrium points. One with high throughput and the other with low throughput caused by collisions during contention. When the system is characterized by this bistability, the iterative technique discussed later in this paper computes the steady state distribution corresponding to the high throughput equilibrium point.

2.2.3. Two-cell stack [16]

Each terminal uses a counter, $\boldsymbol{r}$, as follows.

1. At the start of every request interval the contending terminals initialize their counter, $\boldsymbol{r}$, to 0 or 1 with probability $1/2$.

2. Contending terminals with $\boldsymbol{r} = 0$ transmit into the first request slot. With x being the feedback for that transmission, the transitions in time of $\boldsymbol{r}$ are as follows:
   a. if $x =$ non-collision:
      if $\boldsymbol{r} = 0$, the request packet was transmitted successfully.
      if $\boldsymbol{r} = 1$, then $\boldsymbol{r} = 0$.
   b. if $x =$ collision:
      if $\boldsymbol{r} = 0$, then reinitialize $\boldsymbol{r}$ to 0 or 1 each with probability $1/2$.
      if $\boldsymbol{r} = 1$, then $\boldsymbol{r} = 1$.
3. Repeat step 2, until either two consecutive feedbacks indicating non-collision occur or the request interval ends.

The operation of this protocol can be depicted by a two-cell stack, where in a given request mini-slot the bottom cell contains the transmitting terminals (those with $\boldsymbol{r} = 0$), and the top cell contains the withholding terminals (those with $\boldsymbol{r} = 1$). Although not exploited during voice access, an attractive feature of this algorithm is that two consecutive "non-collisions" indicate that the stack is empty.

2.3. Random access algorithm for data terminals

To transmit data request packets, the data terminals follow the Two-cell stack random access algorithm during consecutive data request intervals. This algorithm was selected because of its operational simplicity, stability and relatively high throughput [16]. Throughout this paper, we will use the term message to differentiate between information and request packets; and, to imply that an arrival event at a data terminal may result in the formation of several information packets.

A blocked access mechanism is established by the following first time transmission rule for newly generated data messages. Terminals with new message arrivals may not transmit during a collision resolution period (CRP). A CRP is defined as the interval of time that begins with an initial collision (if any) and ends with the successful transmission of all data request packets involved in that collision (or, if no collision occurred, ends with that mini-slot). In the first mini-slot following a CRP, all of the terminals whose message arrived within a prescribed allocation interval, of maximum length Δ, transmit with probability one. Terminals involved in a collision follow rules 2–3 in section 2.2.3 and the conclusion of the CRP is identified by two consecutive feedbacks indicating non-collision.

For this algorithm, the maximum data throughput, λ^*, of 0.429 packets per slot is achieved by using a maximum allocation interval, Δ^*, of 2.33 slots [16]. In our simulations, because $\lambda^*\Delta^* \approx 1$, we calculated Δ as $\Delta = 1/\lambda$.

3. Voice traffic analysis

3.1. Assumptions

The primary assumptions used to formulate the approximate Markovian model for voice traffic are as follows.

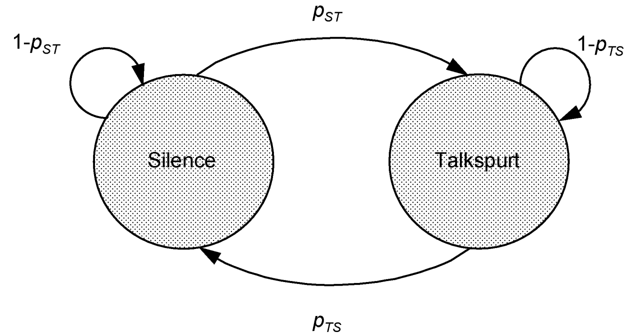


Figure 2. The voice source activity model.

(1) Voice terminals are equipped with a slow voice activity detector [6,17] that only responds to the principal talkspurts (e.g., talk periods > 20 ms) and gaps due to listening and pausing (e.g., silence periods > 200 ms). Thus, voice sources alternate between periods of talkspurt and silence (on and off) and the speech activity is modeled with a two-state discrete time Markov chain as shown in figure 2. The talkspurt to silence transition probability is p_{TS}, and the silence to talkspurt transition probability is p_{ST}. The talkspurt and silence periods are geometrically distributed with mean $1/p_{\mathrm{TS}}$ and $1/p_{\mathrm{ST}}$ frames, respectively. Therefore, at steady state, the probability that a terminal is in talkspurt (speech activity), p_{T}, or silence, p_{S}, is obtained from the following equations:

$$p_{\mathrm{T}} = \frac{p_{\mathrm{ST}}}{(p_{\mathrm{ST}} + p_{\mathrm{TS}})}, \qquad p_{\mathrm{S}} = 1 - p_{\mathrm{T}}. \tag{1}$$

(2) The number of active voice terminals, N, within the system is constant, over the period of interest. This is because the changes in the number of calls is usually on the order of tens of seconds, while the frame duration is on the order of tens of milliseconds [18].

(3) All of the voice transitions (e.g., talk to silence) occur at the frame boundaries.

(4) The voice delay limit, D_{max}, is equal to the duration of two frames. Assumptions (3) and (4), together with the fact that the earliest a contending terminal can receive a reservation is in the frame following the successful transmission of a request, means that a contending voice terminal that fails to successfully transmit a request packet during the voice request interval will drop one voice packet.

(5) The channel is error-free and without capture. Additionally, the base station correctly broadcasts the pending resource allocations before the start of the next frame. As a result, errors within the system only occur when two or more packets arrive simultaneously (collide) at the base station during a request slot.

(6) Reserved slots are deallocated immediately. This implies that a terminal holding a reservation signals the base station upon the completion of a talkspurt. Delayed deallocation schemes can be analyzed with minor

modifications to the method presented below. In a delayed deallocation scheme, the base station determines the end of a talkspurt by detecting silence in a reserved slot (i.e., one information slot is wasted per voice talkspurt).

3.2. System state transitions

As shown in figure 3, we will describe an active voice terminal as being in one of four states: silent, contender, queued, or reserved. A silent terminal has no packets to transmit and does not require channel resources. On transition into talkspurt, the terminal enters the contender state and remains there until it either successfully transmits a request packet or exits talkspurt. Since the requests are queued at the base station, the terminal enters the queued state and remains there until it either receives a reservation (at a minimum, this will be the end of the frame in which the request was received by the base station) or exits talkspurt. After receiving a reservation, the terminal enters the reserved state and transmits one voice packet per frame into its allotted slot until it exhausts its packets and returns to the silent state.

We define the following random variables just before and just after the start of frame k, $k > 0$, with the superscript denoting before:

- T_k^-, T_k, the number of terminals in talkspurt.
- C_k^-, C_k, the number of terminals in the contender state.
- Q_k^-, Q_k, the number of terminals in the queued state.
- R_k^-, R_k, the number of terminals in the reserved state.

Given N active voice terminals in the system, the number of terminals in the silent state, just before the start of frame k, is equal to $N - T_k^-$, and $T_k^- = C_k^- + Q_k^- + R_k^-$. Thus, the state of the system is completely described by three random variables, such as T_k^-, C_k^-, and R_k^-. Unfortunately, the size of the state space grows very large for any interesting system.

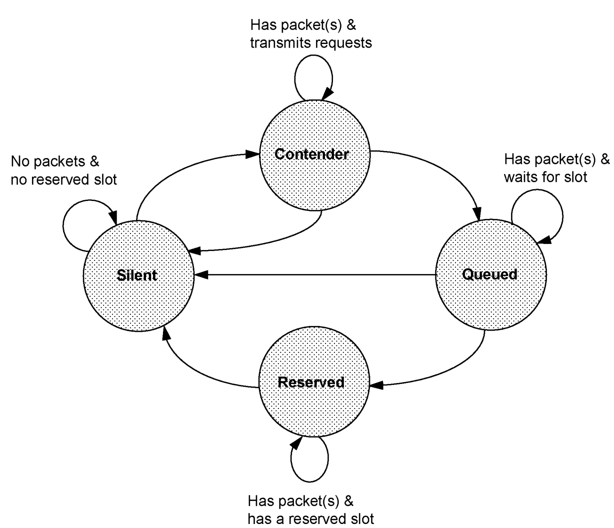


Figure 3. State transition diagram for an active voice terminal.

Recall that the base station allocates reservations at the end of each frame. Thus, when the system operates under reasonable average voice packet dropping probability constraints, the Q_k^- term will often be subsumed by the R_k^- term. We can make our model tractable by assuming that the base station allocates resources according to the transitions that occur at the frame boundaries (i.e., if a reserved terminal transitions to silence, the base station may allocate the reserved terminal's slot). Thus, we let $X_k^- = Q_k^- + R_k^-$, and $X_k = Q_k + R_k$. Then, since $T_k^- - C_k^- = X_k^-$, the state of the system can be described by the random variables T_k^- and C_k^-; and, we can use aspects of the methodology in [9] to formulate the Markovian process $Z^- = \{Z_k^- = (T_k^-, C_k^-),\ k > 0\}$.

According to our allocation policy, a voice terminal in the queued state will receive a reservation for the next frame unless every information slot has already been allocated to voice terminals (the frame is full). Thus, knowing the value of X_k^-, we obtain the values of Q_k^- and R_k^- as follows:

$$\begin{aligned}&\text{if } X_k^- \leqslant I, \text{ then} \quad R_k^- = X_k^- \quad \text{and} \quad Q_k^- = 0,\\ &\text{if } X_k^- > I, \text{ then} \quad R_k^- = I \quad \text{and} \quad Q_k^- = X_k^- - I,\end{aligned} \tag{2}$$

where I is the number of information slots per frame.

The assumption that the voice transitions occur only at frame boundaries allows us to separate the voice transitions (terminals entering/exiting talkspurt) from the random access (contention) process [9]. Thus, the Markov process, Z^-, can be viewed as evolving in two steps. The following equations account for the voice transitions:

$$\begin{aligned}T_k &= T_k^- + A_k - S_k^C - S_k^X,\\ C_k &= C_k^- + A_k - S_k^C,\end{aligned} \tag{3a}$$

where A_k is the number of terminals entering talkspurt at the beginning of frame k, S_k^C is the number of contending terminals exiting talkspurt during frame k, and S_k^X is the number of queued and/or reserved terminals exiting talkspurt during frame k.

Separating the transitions from the contention means that $T_{k+1}^- = T_k$. Thus, with W_k being the number of successfully transmitted request packets within the voice request interval of frame k, the following equation accounts for the outcome of the contention:

$$C_{k+1}^- = C_k - W_k. \tag{3b}$$

3.2.1. Ideal random access

According to this protocol, each contending voice terminal in frame k successfully transmits its request within the voice request interval of frame k. Thus, for all $k > 0$, $C_k = A_k = W_k$ and $C_{k+1}^- = 0$. And, since T_k includes A_k, we only need to consider the process

$$Y^- = \{Y_k^- = (T_k^-),\ k > 0\}.$$

Y^- evolves according to the recurrence equation $T_{k+1}^- = T_k^- + A_k - S_k^X$, and its limiting distribution can be derived in a way similar to below. But, since the steady state distribution of Y^- corresponds to the probability that m out of N terminals are in talkspurt, P_m, it depends only on the speech activity. Because we model speech activity with a two-state discrete time Markov chain, the steady state distribution of Y^- is readily calculated as follows:

$$\Pr\{Y^- = m\} = P_m = \mathrm{B}(N, m, p_{\mathrm{T}}), \tag{4}$$

where $\mathrm{B}(n, m, p) = \binom{n}{m} p^m (1-p)^{n-m}$ is the binomial distribution.

3.2.2. Slotted Aloha and two-cell stack random access

In both the slotted Aloha and the two-cell stack algorithms, a contending voice terminal may fail to successfully transmit a request packet within the voice request interval (e.g., due to collisions). In that case, the voice terminal attempts to transmit requests in succeeding frames, until it either succeeds or exits talkspurt. Thus, for all $k > 0$, $C_{k+1}^- = C_k - W_k \geqslant 0$ and we are obliged to consider the process

$$Z^- = \{Z_k^- = (T_k^-, C_k^-),\ k > 0\}.$$

The evaluation W_k requires the probability distribution, $F_{r,c}(w) = \Pr\{$exactly w out of c contenders succeed within the voice interval (r mini-slots) of the frame$\}$. Clearly, $F_{r,c}(w)$ depends on the random access protocol used. Although the general methodology used to calculate $F_{r,c}(w)$ for both of these random access protocols can be found in [9], we have included the specific expressions in appendices A and B for completeness.

With N active voice terminals in the system and r mini-slots in the voice request interval, the one step transition probability, $p_{mn,ij}$, is expressed as follows:

$$\begin{aligned}
&\Pr\{T_{k+1}^- = i,\ C_{k+1}^- = j \mid T_k^- = m,\ C_k^- = n\} \\
&\quad = \sum_{h=j}^{\min(N-m+n, r+j)} \Pr\{T_k = i,\ C_k = h \mid T_k^- = m,\ C_k^- = n\} \\
&\qquad \times \Pr\{T_{k+1}^- = i,\ C_{k+1}^- = j \mid T_k = i,\ C_k = h\}.
\end{aligned} \tag{5}$$

The first term on the right hand side of (6) accounts for the voice transitions and it can be calculated by conditioning on A_k as follows:

$$\begin{aligned}
&\Pr\{T_k = i,\ C_k = h \mid T_k^- = m,\ C_k^- = n\} \\
&\quad = \sum_{a=\max(h-n,0)}^{\min(N-m,h)} \sum_{a=\max(h-n,0)}^{\min(N-m,h)} \Pr\{T_k = i,\ C_k = h \mid T_k^- = m,\ C_k^- = n,\ A_k = a\} \\
&\qquad \times \Pr\{A_k = a \mid T_k^- = m,\ C_k^- = n\} \\
&\quad = \sum_{a=\max(h-n,0)}^{\min(N-m,h)} \mathrm{B}(n, a+n-h, p_{\mathrm{TS}}) \\
&\qquad \times \mathrm{B}(m-n, m-n-i+h, p_{\mathrm{TS}}) \\
&\qquad \times \mathrm{B}(N-m, a, p_{\mathrm{ST}}),
\end{aligned} \tag{6}$$

where $a+n-h$ and $m-n-i+h$ are equal to the number of contender and queued/reserved terminals that exit talkspurt, respectively.

The second term on the right hand side of equation (6) accounts for the contention and it is expressed as follows:

$$\begin{aligned}
&\Pr\{T_{k+1}^- = i,\ C_{k+1}^- = j \mid T_k = i,\ C_k = h\} \\
&\quad = F_{r,c}(h-j).
\end{aligned} \tag{7}$$

Notice that the conditioning on T_k^-, C_k^- is dropped; since given T_k and C_k, T_{k+1}^- and C_{k+1}^- are independent of T_k^- and C_k^- [9].

3.3. Voice performance measures

We limit our discussion to the process

$$Z^- = \{Z_k^- = (T_k^-, C_k^-),\ k > 0\},$$

since the application to Y^- is straightforward. With an eye towards formulating expressions for the steady state voice packet dropping probability, mean access delay and throughput, we let $\mathrm{E}[T^-]$, $\mathrm{E}[C^-]$, $\mathrm{E}[Q^-]$ and $\mathrm{E}[R^-]$ be the steady state mean number of voice terminals in the talkspurt, the contender, the queued and the reserved state per frame, respectively. Since $\mathrm{E}[T^-]$ depends only on the speech activity, it is equal to the mean of equation (4):

$$\mathrm{E}[T^-] = N p_{\mathrm{T}}. \tag{8}$$

Given the steady state distribution of Z^-, $\pi(i,j)$, we use the following equations to evaluate $\mathrm{E}[C^-]$ and $\mathrm{E}[Q^-]$:

$$\mathrm{E}[C^-] = \sum_{i=0}^{N} \sum_{j=0}^{i} j \pi(T_k^- = i,\ C_k^- = j), \tag{9}$$

$$\begin{aligned}
\mathrm{E}[Q^-] = &\sum_{i=0}^{N} \sum_{j=0}^{\max(i-j-I,0)} (i-j-I) \\
&\times \pi(T_k^- = i,\ C_k^- = j),
\end{aligned} \tag{10}$$

where I is the number of information slots per frame. Then, by definition, $\mathrm{E}[R^-]$ is obtained as follows:

$$\mathrm{E}[R^-] = \mathrm{E}[T^-] - \mathrm{E}[C^-] - \mathrm{E}[Q^-]. \tag{11}$$

3.3.1. Packet dropping probability

The steady state voice packet dropping probability is the ratio of the average number of voice packets dropped per frame to the average number of voice packets generated per frame. By our assumptions, every voice terminal that is contending and every voice terminal with a packet in the queue at the end of the frame (i.e., all information slots are reserved by voice terminals) will drop one packet. We let P_{drop} represent the steady state voice packet dropping probability percent, and calculate it as follows:

$$P_{\mathrm{drop}} = 100 \frac{\mathrm{E}[C^-] + \mathrm{E}[Q^-]}{\mathrm{E}[T^-]}. \tag{12}$$

3.3.2. Voice access delay

The access delay for a voice terminal is the time between the start of a talkspurt and the end of the first voice packet transmission in its reserved slot. For the RRA scheme under investigation, the mean access delay, D, can be expressed as

$$D = D_c + D_q + D_r, \tag{13}$$

where D_c is the mean random access delay; D_q is the mean queuing delay; and, D_r is the mean time between the start of the frame in which the reservation is granted and the end of the transmission in its reserved slot. Recall that upon successful receipt of a request packet, the base station queues the request until it allocates new reservations (if available) at the end of the frame. To simplify the following presentation, we express time in information slots and we define the minimum value of D_c to be one information slot (i.e., the entire voice interval). For example, a terminal that successfully transmits a request during the voice request interval and receives a reservation for the next frame will experience: $D_c = 1$ slot; $D_q = F - 1$ slots, where, F is the number of slots per frame; and, D_r such that $2 < D_r \leqslant F$ slots (i.e., 2 slot delay for the request intervals plus a delay equal to some number of information slots).

Recall that a terminal will be in one of four states: silent, contender, queued or reserved. Under steady state conditions, the input and output of each state must be equal. Thus, the mean delay due to terminals that fail to successfully transmit a request during the frame is the ratio of the average number of terminals in the contender state to the average rate of terminals entering the contender state (Little's result [4]). The number of terminals entering the contender state is equal to the number of terminals entering talkspurt. At steady state, the expected number of voice terminals entering talkspurt at the start of a frame is equal to the product of the mean number of silent terminals (i.e., Np_{S}) and the silence to talkspurt transition probability, p_{ST}. Thus, D_c is obtained as follows:

$$D_c \approx 1 + F\frac{\mathrm{E}[C^-]}{Np_{\mathrm{S}}p_{\mathrm{ST}}}. \tag{14}$$

When all of the slots within the frame are reserved, newly arriving request packets are queued until resources become available. This delay, on average, is equal to the ratio of the number of terminals in the queued state to the number of terminals entering the queued state (Little's result [4]). Therefore, the following equation is used to estimate D_q:

$$D_q \approx F - 1 + F\frac{\mathrm{E}[Q^-]}{Np_{\mathrm{S}}p_{\mathrm{ST}}}. \tag{15}$$

Recall our assumption that the base station always allocates the earliest empty information slot within the frame. Due to independent talk/silence transitions, terminals with reservations will exit talkspurt randomly. Thus, by assuming that the reserved slot is uniformly distributed over the interval between the first information slot and the number of reserved slots in the frame, D_r is approximated as follows:

$$D_r \approx 2 + \sum_{i=0}^{N}\sum_{j=0}^{i}\bigl(0.5\ g(i,j) + 1\bigr) \times \pi(T_k^- = i,\ C_k^- = j), \tag{16}$$

where

$$g(i,j) = \begin{cases} i-j, & i-j \leqslant I, \\ I, & \text{otherwise.} \end{cases}$$

3.3.3. Voice throughput

Throughput is defined as the proportion of time slots that successfully carry information packets from terminals to the base station. Thus, we can express the voice throughput, η, as the mean number of successfully transmitted voice packets per frame. Recall that a contending terminal in frame k is not eligible to receive a reservation until frame $k+1$. As a result, terminals that are in the contender and queued states do not transmit a voice packet during the frame. Our assumption of an error-free channel means that every terminal in the reserved state at the end of the frame succeeded in transmitting its voice packet during the frame. Therefore, $\eta = \mathrm{E}[R^-]$.

4. Performance evaluation

We employ analysis and simulations to study the proposed RRA scheme together with the random access algorithms described in section 2. To further characterize the voice traffic quality, simulations are used to investigate the distribution of the voice packets dropped per talkspurt. Finally, we employ simulations to explore preliminary voice-data integration issues.

4.1. System parameters

We use the parameters contained in table 1 to perform the analysis and the simulations. The channel rate is from [11,12]. The speech rate value assumes the use of 32 kb/s adaptive differential pulse code modulation (ADPCM) in the microcellular environment [1]. The packet

Table 1
Experimental system parameters.

Parameter	Value
Channel rate	1.8 Mb/s
Speech rate	32 kb/s
Information packet: (header/payload)	424 b (40/384 b)
Frame duration	12 ms (50 slots)
Voice delay limit	24 ms
Voice request interval	1 slot/frame
Data request interval	1 slot/frame
Information interval	48 slots/frame
Mini-slots per request interval	6
Mean talkspurt duration	1.41 s
Mean silence duration	1.78 s

size (53 bytes) was selected for compatibility with ATM networks. The values for the mean talkspurt/silence duration produce a speech activity of approximately 44 percent [11]. It is important to note that other conversation activities such as listening to voice mail or speaking to an answering machine lead to different talkspurt/silence distributions. However, we selected the numbers used in this study to be consistent with others that consider slow voice activity detection; for example, the mean talkspurt/silence duration values of 1.0/1.35 s [6] and 1.41/1.74 s [12] produce speech activities of approximately 43 and 45 percent, respectively.

Given the above parameters, the frame duration is 12 ms (i.e., payload/speech rate) thereby accommodating approximately 51 slots (i.e., information slots). The voice delay limit of 24 ms is equal to the duration of two frames (i.e., $D_{\max} = 24$ ms). To account for guard time and synchronization, we assume that 50 slots are available per frame. The first and second slots are dedicated to the voice and data request intervals, respectively, and are both subdivided into mini-slots. The remaining 48 slots comprise the information interval. We chose six mini-slots per request interval (so that each mini-slot accommodates the transmission of approximately 70 bits) to allow for guard time and synchronization overheads and for the transmission of a generic request packet that contains the source identifier, along with some data (e.g., priority, slots required, etc.).

4.2. Calculation of the limiting distribution for Z^-

The process Z^- is ergodic, because it is an irreducible, aperiodic Markov chain, with finite state space. Therefore, with P being the one step transition probability matrix, the limiting distribution, π satisfies the following equations:

$$\pi = \pi P, \tag{17}$$

$$\sum_{i=0}^{s} \pi_i = 1, \tag{18}$$

where S is the size of the state space. For Z^-, given N active voice terminals, $S = (N+1)(N+2)/2$. An iterative solution to (17) is possible; since, with $\pi(0)$ being any arbitrary distribution, the limiting distribution of a Markov chain may be expressed as follows:

$$\pi = \lim_{n\to\infty} \pi(0)P^n. \tag{19}$$

The iteration is stopped when, from one iteration to the next, each element in the distribution meets the criterion

$$\left|\pi_i(n+1) - \pi_i(n)\right| \leqslant \varepsilon, \quad 0 \leqslant i \leqslant S, \tag{20}$$

where ε is some small value. A scaling operation was used to ensure that the elements of the distribution summed to 1 after each iteration [19].

With the parameters contained in table 1, the system accommodates approximately 100 active voice terminals and, for Z^-, S is over 5000 states. The limiting distribution can be obtained via iteration with a desktop workstation; and, a considered choice of $\pi(0)$ makes convergence very rapid (approximately 5 iterations with $\varepsilon = 10^{-6}$). For the process Z^-, we observe that a good random access algorithm is one that permits contending terminals to successfully transmit request packets (i.e., one that minimizes C_k^-). In the best case (e.g., the ideal algorithm), every contending terminal would successfully transmit its request during the voice request interval and $C_k^- = 0$. Therefore, for $0 \leqslant m \leqslant N$, we construct $\pi(0)$ as follows. For all of the states with $T_k^- = m$ and $C_k^- = 0$ set $\pi(0) = P_m$, as calculated by equation (4); and, for all of the states with $T_k^- = m$ and $C_k^- > 0$ set $\pi(0) = 0$.

4.3. Simulations

All simulations consist of ten independent runs of 305,000 frames each. To reduce start up effects, the first 5000 frames serve as the warm up period. During each run: the number of terminals within the system is held constant; the terminals are initially silent; and, the results for specific performance measures are calculated with cumulative data. For example, the steady state voice packet dropping probability is obtained from the ratio of the total number of voice packets dropped to the total number of voice packets generated over the simulation run.

5. Results and discussion

5.1. Voice traffic

Analytical and simulation results for voice performance measures, such as P_{drop} vs. the number of active voice terminals, N, for each of the access protocols are provided in figures 4–8. In every figure, both the analysis and the simulation curves are generated from results obtained with even values of N (i.e., $N = 70$, 72, 74, etc.). The analytic results contained in tables 2 and 3 were obtained with $N = 97$.

For the system under investigation, the steady state P_{drop}, mean access delay and throughput increase with N. A common measure of comparison for RRA voice protocols operating in a microcellular environment is the maximum N with $P_{\text{drop}} \leqslant 1\%$ (voice capacity). The analytical results obtained for P_{drop}, voice throughput, and mean voice access delay for the random access protocols operating at their respective voice capacity are summarized in table 2. The multiplexing gain is the ratio of the voice capacity to the

Table 2
Steady state voice performance at voice capacity.

Access protocol	N (terminals)	P_{drop} (percent)	Mean access delay (ms)	Throughput (packet/frame)	Multiplexing gain
Aloha	97	0.963	31.40	42.46	1.94
Two-cell	97	0.905	30.57	42.49	1.94
Ideal	97	0.868	30.07	42.50	1.94

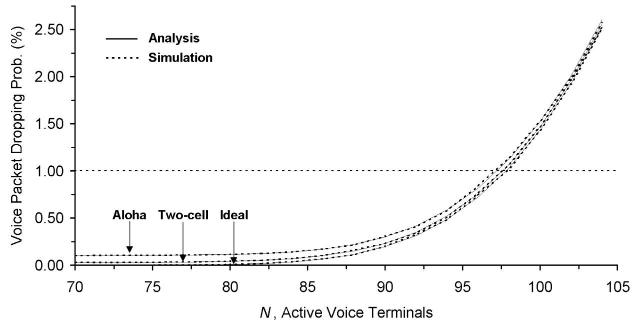


Figure 4. Steady state voice P_{drop} vs. N.

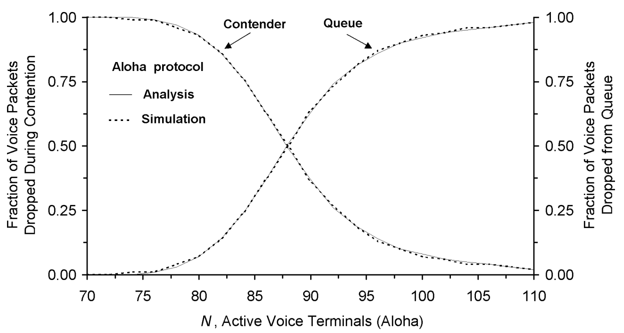


Figure 5. Fraction of voice packets dropped from the contender and queued states; slotted Aloha.

number of slots per frame (i.e., N/F). For a stable system, without a P_{drop} constraint, the optimal gain for this system is 2.16 (i.e., $I/(Fp_{\text{T}})$, where I is the number of information slots/frame). Without a P_{drop} constraint larger values of N are possible, however as the input rate to the queue grows larger than the achievable output rate (I packets/frame) the system will experience instability even with an ideal scheduler.

From figure 4 we observe that for N values below about 97 the Aloha access algorithm consistently yields slightly higher P_{drop} values than the two-cell and ideal algorithms; and, that for $N > 97$ all of the algorithms produce similar P_{drop} values. We also see that the analytical results agree with those obtained via simulation.

Voice packet dropping is due to delays encountered by a terminal that is in either the contender (random access) or queued state (waiting time longer than $F-1$ slots, when all of the information slots are allocated to voice terminals), since we assume that every talk/silence transition occurs at the frame boundaries. Thus, it is worth investigating the fraction of voice packets dropped from terminals in the contender and queued states. The steady state number of voice packets dropped per frame from terminals in the contender and queued states were obtained analytically using the definitions in section 3. For the simulations, a state variable is included with each terminal and the packets dropped were tracked accordingly. The results obtained for the Aloha and two-cell algorithms are shown in figures 5 and 6, respectively. With ideal access, all voice packet dropping occurs in the queued state.

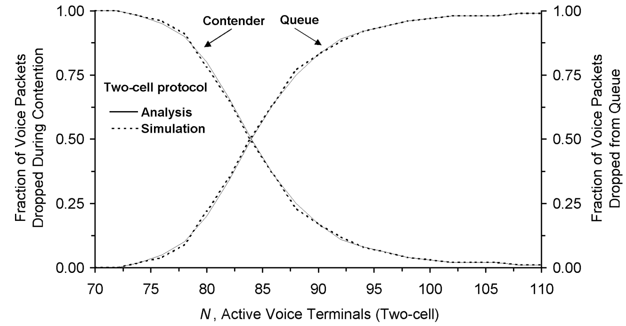


Figure 6. Fraction of voice packets dropped from the contender and queued states; two-cell stack.

The contender and queued lines intersect at about $N = 88$ and $N = 84$ for Aloha (figure 5) and the two-cell (figure 6) access algorithms, respectively. This indicates that packet dropping due to contention is more significant for Aloha than for the two-cell algorithm. We explain this by noting that, at steady state, the expected number of voice terminals entering talkspurt at the start of a frame is equal to the product of the mean number of silent terminals and the silence/talkspurt transition probability (i.e., $Np_{\text{S}}p_{\text{ST}}$). Thus, the expected number of new contenders per frame is much less than one (e.g., from 0.26 to 0.38, for N values from 70 to 100). When exactly one contending terminal follows the Aloha algorithm in section 2, it successfully transmits its request 90 percent of the time, due to the probabilistic first time transmission rule and the use of six mini-slots per voice request interval. However, when exactly one contending voice terminal follows the two-cell algorithm it always succeeds. Therefore, from the results in figures 4–6, as expected the packet dropping depends on the random access algorithm at lower loads (i.e., approximately, $N < 82$ or gain < 1.64). At high loads where dropping from the queuing delay is predominant, although the choice of random access algorithm does not improve the voice capacity (or significantly improve the throughput) it does improve the P_{drop} and mean access delay.

It is well known that the steady state voice packet dropping probability may not be an accurate indicator of voice performance when considering quality of service issues or access control strategies. For example, dropped packets result in front end clipping which may significantly degrade the quality of voice conversations. Table 3 contains simulation results for the steady state packet dropping distribution per talkspurt for each access protocol operating at voice capacity.

At voice capacity, the P_{drop} values obtained via simulation using the Aloha, two-cell, and ideal random access algorithms are equal to 0.970, 0.919 and 0.863, respectively. Comparing the Aloha values to the two-cell values in table 3, suggests that the probability of dropping zero or one voice packet per talkspurt is largely due to the efficiency of the random access algorithm; while, dropping more than one is due to the frame being full. The probability of dropping more than one voice packet is 0.14, 0.14 and 0.12 when the Aloha, two-cell and ideal access

Table 3
Steady state distribution of the voice packets dropped per talkspurt at voice capacity.

Packets dropped per talkspurt	Aloha $N = 97$	Two-cell $N = 97$	Ideal $N = 97$
0	0.76	0.82	0.86
1	0.10	0.04	0.02
2	0.03	0.02	0.02
3	0.01	0.02	0.01
4	0.01	0.01	0.01
5	0.01	0.01	0.01
6	0.01	0.01	0.01
7	0.01	0.01	0.01
8	0.01	0.01	0.01
9	0.01	0.01	0.01
> 9	0.04	0.04	0.03

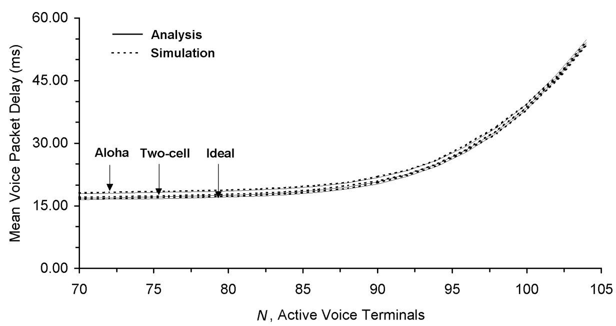


Figure 7. Steady state mean voice access delay.

protocols are used, respectively. In addition, the results in table 3 indicate that an arriving talkspurt has a three or four percent chance of dropping more than nine packets before receiving a reservation (or exiting talkspurt). This possibility of an extended wait in the queued state is a consequence of the average talkspurt duration being on the order of 100 frames. Although not included in the table, our simulations indicate a non-zero probability for run lengths of up to approximately 53 dropped packets (almost one half of the average talkspurt length).

As seen in figure 7, the steady state mean voice access delay curves for the three access protocols show similar trends to those for P_{drop} vs. N, in figure 4; and, for the same reasons as above, the Aloha access algorithm consistently produces slightly higher mean access delay values up until about $N = 97$. Additionally, we observe that the mean access delay increases above the voice limit delay of 24 ms when $N(P_{\text{drop}})$ is approximately equal to 92 (0.42), 94 (0.52) and 94 (0.49) for the Aloha, two-cell and ideal access protocols, respectively. These values indicate that the queuing delay component unduly influences the steady state mean access delay value.

Figure 8 shows that the voice throughput, η, is nearly linear until P_{drop} begins to exceed one percent (about $N = 98$), regardless of the access protocol used. Clearly, the voice throughput depends on the speech activity and the packet dropping probability. With our notation, η, is accurately

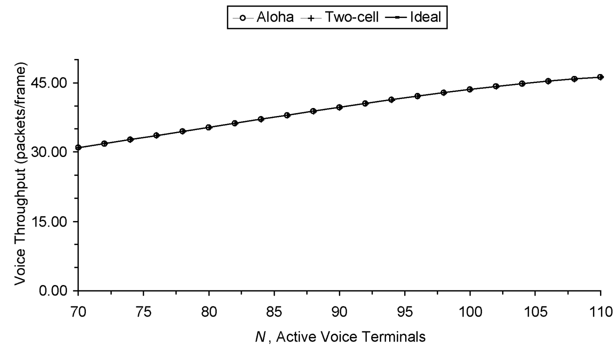


Figure 8. Analytical results for voice packet throughput vs. N.

described by the following equation [9]:

$$\eta = N p_{\mathrm{T}} \left(1 - \frac{P_{\text{drop}}}{100} \right). \tag{21}$$

5.2. Voice-data integration

Although we have focused mainly on the ability of the RRA scheme to support voice traffic, we recognize that data traffic will gain in importance as small, portable, and inexpensive computing devices proliferate. One difficulty in designing an algorithm for voice-data integration is determining how to characterize and model the user side data traffic in the emerging wireless environment. Besides assuming that voice is of higher priority than data traffic, our proposed RRA scheme is based on the following considerations. First, we assume that data applications with special needs such as high speed or minimal variance between packet deliveries will be accommodated with reservations and a suitable priority designation. Next, we surmise that a typical PCS data user might do things like read and compose short e-mail messages, respond to paging type messages, transfer work (files) between the portable device and a fixed computer at home/office, and query some kind of database. Finally, we observe that the data may consist of short mouse events or text-based characters.

As described in section 2, a data terminal that successfully transmits a request packet receives a reservation, but, since it transmits low priority traffic, its reservation may be preempted to service a voice terminal. Thus, we consider the following performance measures as important indicators for the data traffic. The wait delay (or access delay) is defined as the time between the message arrival and the end of the first data packet transmission into a reserved slot. The message delay is defined as the time between the message arrival and the end of the last data packet transmission into a reserved slot. And, the throughput is defined as the proportion of time slots that successfully carry data information packets.

We assume that data messages are generated by a large unknown number of data terminals (theoretically infinite) and that the aggregate message arrivals are Poisson distributed with mean λ messages per frame. Additionally, we assume that the messages vary in length according to a

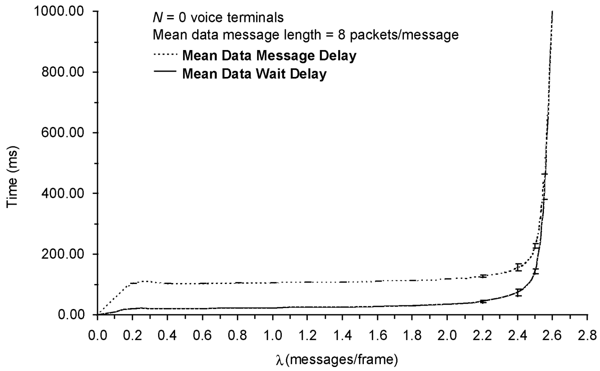


Figure 9. Steady state mean data delays; $N = 0$.

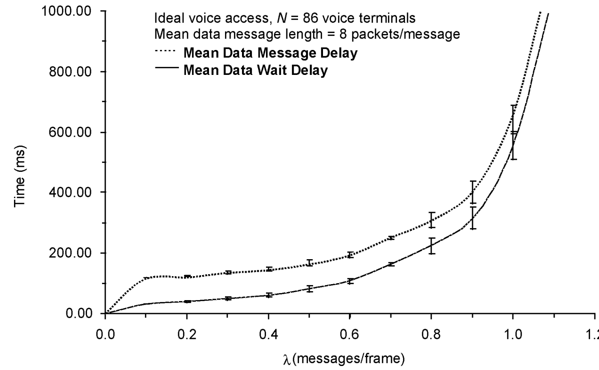


Figure 10. Steady state mean data delays; $N = 86$.

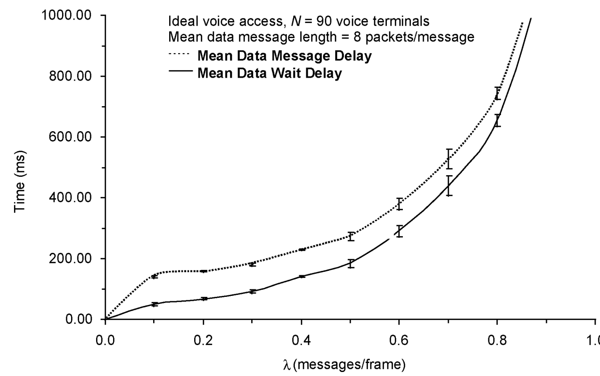


Figure 11. Steady state mean data delays; $N = 90$.

geometric distribution with parameter q and mean $B = 1/q$. Because the duration of an information slot is fixed (and fairly short), an arriving data message may result in the formation of multiple packets. Thus, expressing B as packets per message, the steady state data rate is equal to λB packets/frame. We recognize that data traffic may be more "bursty" than our traffic model indicates, but the model's common use makes it reasonable for initial investigations.

For the results shown in figures 9–11, the parameter q is always equal to $1/8$ (thus, $B = 8$); and, the error bars denote the 97.5 % t-confidence intervals (constructed in the usual way [20]). Our choice of q implies that the average data message is short because it only contains approximately 3400 bits, and because each packet will contain overhead bits for forward error correction plus some overhead bits from the upper layers.

Simulation results for the data wait delay and the data message delay vs. the data message arrival rate, λ, for the system with $N = 0$ (no voice traffic), are shown in figure 9. We observe that the data message delay is consistently greater than the data wait delay by about 84 ms. This is because, unless a data reservation is preempted, the message delay equals the sum of the wait delay and $(B - 1)f$, where f is the frame duration (ms). Additionally, we see that the data message delay is below 200 ms, until λ equals about 2.5 messages/frame, then it increases sharply (about 400 ms for $\lambda = 2.55$ and off the scale for $\lambda = 2.6$). We explain this result by observing that the maximum throughput of the two-cell stack algorithm (0.429 [16]) together with six mini-slots per frame means that the maximum $\lambda = 0.429 \times 6 = 2.575$ data messages/frame. Since we assumed that data terminals only transmit one packet per frame, the maximum data packet throughput is about 20.6 packets per frame.

The fact that voice and data terminals transmit their request packets during separate request intervals together with a channel resource allocation policy that preempts data in favor of voice means that the addition of data traffic will not affect the voice traffic performance. There are two costs to this approach: the fixed overhead due to dedicating a portion of the frame to data requests; and, at high voice loading, data throughput is severely limited and the data wait and message delays increase significantly. For example, from table 2, the ideal protocol operating at its steady state voice capacity provides voice throughput equal to 42.5 packets/frame with an access delay of about 30 ms. Thus, the maximum data throughput is limited to an average of about 5.5 packets/frame (i.e., less than the mean message length that we have assumed here); and, it suggests that the data wait and message delays will be intolerable due to the real possibility that all of the information slots within the frame will be allocated to voice terminals. Therefore, we investigate the data traffic performance under slightly less severe voice loading.

In figures 10 and 11 we provide the simulation results obtained for the mean data delays vs. λ when voice terminals employ the ideal access algorithm. In figure 10, $N = 86$ voice terminals and steady the state voice P_{drop}, mean access delay and throughput are approximately 0.07, 18 ms and 38 packets/frame, respectively. When $N = 90$ voice terminals (figure 11), the steady state voice P_{drop}, mean access delay and throughput are about 0.2, 20 ms and 40 packets/frame, respectively. Since, with ideal voice access, the P_{drop} values are due to the frame being full, there will be consecutive frames when voice traffic inhibits the allocation of data reservations. The results suggest that for $N = 86$ (90), the average data message delays are below 200 ms for message arrival rates up until about 0.65 (0.40) messages per frame. In both cases, the total system load is approximately 86 percent.

6. Conclusion

We have proposed and analyzed an RRA scheme for integrating voice and data traffic in outdoor microcellular environments. The scheme involved partitioning a portion of the frame into mini-slots, along the general lines of [13,14]. We formulated an approximate Markov model and derived expressions for the steady state voice performance metrics of packet dropping probability, mean access delay and throughput. Analytical results were provided to illustrate the voice performance of several multiple access algorithms. Additionally, simulation results were used to verify the analysis, to study the steady state voice packet dropping distribution per talkspurt and to explore preliminary voice-data integration considerations. Our results suggest that this is a promising scheme for providing voice-data integration in outdoor microcellular environments.

Appendix A. Evaluation of $F_{r,c}(w)$ for slotted Aloha [9]

Given c contending voice terminals at the start of a request slot, each one will transmit in the slot with probability p. Thus, the probability that one of the terminals will transmit successfully, $P_{\mathrm{s}}(c)$, is calculated as follows:

$$P_{\mathrm{s}}(c) = \begin{cases} cp(1-p)^{c-1}, & c > 0, \\ 0, & c = 0. \end{cases}$$

The probability, $F_{r,c}(w)$, that exactly w terminals successfully transmit their request within the r slots of the voice request interval, given that there are c contending terminals when the frame starts, is obtained from the following recursion:

$$F_{r,c}(0) = \left(1 - P_{\mathrm{s}}(c)\right)^r,$$

$$F_{r,c}(w) = \begin{cases} P_{\mathrm{s}}(c)F_{r-1,c-1}(w-1) + (1-P_{\mathrm{s}}(c))F_{r-1,c}(w), \\ \quad 0 < w < \min(r,c),\ r > 0,\ c > 0, \\ 0, \quad \text{otherwise}. \end{cases}$$

Appendix B. Evaluation of $F_{r,c}(w)$ for two cell stack [9]

Consideration of the two cell stack algorithm reveals that $F_{r,c}(w)$ can be evaluated recursively. Let $F_{r,c0,c1}(w)$ be the conditional probability of w successes; given r slots in the voice interval, and, $c0$ and $c1$ terminals with their counter values equal to zero and one, respectively. Then, since every contending terminal initializes its counter value to 0 or 1 with equal probability at the start of the voice request interval, we can express $F_{r,c}(w)$ as follows:

$$F_{r,c}(w) = \sum_{i=0}^{c} F_{r,i,c-i}(w)\binom{c}{i}2^{-c}.$$

The following recursive identities hold for $F_{r,c0,c1}(w)$:

$$F_{r,0,c1}(w) = F_{r-1,c1,0}(w),$$
$$F_{r,1,c1}(w) = F_{r-1,c1,0}(w-1),$$
$$F_{r,c0,c1}(w) = F_{r-1,i,c0+c1-i}(w) \quad \text{w.p. } \binom{c0}{i}2^{-c0}.$$

And, the initial conditions are as follows:

$$\begin{aligned} F_{r,c0,c1}(w) &= 0, \quad \text{if } w > r \text{ or } w > c0 + c1, \\ F_{r,1,0}(1) &= 1, \quad \text{if } r > 0, \\ F_{1,c0,c1}(0) &= 1, \quad \text{if } c0 \neq 1, \\ F_{1,c0,c1}(0) &= 0, \quad \text{if } c0 = 1. \end{aligned}$$

References

[1] D.C. Cox, Wireless personal communications: What is it? IEEE Personal Comm. 2(2) (April 1995) 20–35.

[2] V.O.K. Li and X. Qui, Personal communication systems, Proc. IEEE 83(9) (September 1995) 1210–1243.

[3] N. Abramson, Multiple access in wireless digital networks, Proc. IEEE 82(9) (September 1994) 1360–1370.

[4] D. Bertsekas and R. Gallager, *Data Networks* (Prentice-Hall, Englewood Cliffs, NJ, 2nd ed., 1992).

[5] D.J. Goodman and S.X. Wei, Efficiency of packet reservation multiple access, IEEE Trans. Vehicular Technol. 40(1) (February 1991) 170–176.

[6] S. Nanda, D.J. Goodman and U. Timor, Performance of PRMA: A packet voice protocol for cellular systems, IEEE Trans. Vehicular Technol. 40 (August 1991) 584–598.

[7] M. Frullone, G. Riva, P. Grazioso and C. Carciofi, Packet reservation multiple access in cellular environments, European Trans. on Telecomm. 5(1) (January–February 1994) 51–58.

[8] W.C. Wong and D.J. Goodman, Integrated data and speech transmission using packet reservation multiple access, in: *Proc. ICC '93*, Geneva, Switzerland (May 1993) pp. 172–176.

[9] S. Jangi and L. Merakos, Performance analysis of reservation random access protocols for wireless access networks, IEEE Trans. Comm. 42(2–4) (1994) 1223–1234.

[10] A.C. Cleary and M. Paterakis, An investigation of reservation random access algorithms for voice-data integration in microcellular wireless environments, Internat. J. Wireless Info. Networks 2(1) (January 1995) 1–16.

[11] J.M. DeVile, A reservation based multiple access scheme for a future universal mobile telecommunications system, in: *Proc. 7th IEE Conf. on Mobile and Personal Comm.*, Brighton, UK (December 1993) pp. 210–215.

[12] J. Dunlop, J. Irvine, D. Robertson and P. Cosimini, Performance of a statistically multiplexed access mechanism for a TDMA radio interface, IEEE Personal Comm. 2(3) (June 1995) 56–64.

[13] N.M. Mitrou, T.D. Orinos and E.N. Protonotarios, A reservation multiple access protocol for microcellular mobile communication systems, IEEE Trans. Vehicular Technol. 39(4) (August 1990) 340–351.

[14] N.D. Wilson, R. Ganesh, J. Kuriacose and D. Raychaudhuri, Packet CDMA versus dynamic TDMA for multiple access in an integrated voice/data PCN, IEEE J. Selected Areas Comm. 11(6) (August 1993) 870–884.

[15] B. Mukherjee, Integrated voice-data communication over high speed fiber optic networks, IEEE Computer Mag. 24(2) (February 1991) 49–58.

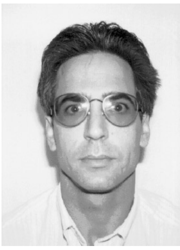
[16] M. Paterakis, L. Georgiadis and P. Papantoni-Kazakos, A full sensing window RA algorithm for networks with strict delay constraints, Algorithmica 4 (1989) 313–328.

[17] P.T. Brady, A technique for investigating on–off patterns of speech, Bell Syst. Tech. J. (January 1965).

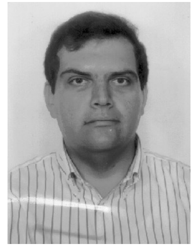
[18] T. Suda and T. Bradley, Packetized voice/data integrated transmission on a token passing ring local area network, IEEE Trans. Comm. 37(3) (March 1989) 238–244.

[19] J.E. Wieselthier and A. Ephremides, A new class of protocols for multiple access in satellite networks, IEEE Trans. Automat. Control 25(5) (October 1980) 865–879.

[20] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis* (McGraw Hill, 2nd ed., 1991).



Allan C. Cleary received the B.S. degree from Clemson University and the M.S. degree from the University of Massachusetts-Lowell, in 1983 and 1985, respectively, both in chemical engineering. In 1992, he received the M.S. degree from the University of Delaware in computer and information sciences. Since 1985 he has been employed as a chemical engineer at the US Army Edgewood Research Development and Engineering Center, Aberdeen Proving Ground, MD. In addition, he is completing the Ph.D. degree in computer and information sciences at the University of Delaware. His research interests include stochastic processes with emphasis on modeling, simulation and performance evaluation of communication networks. He is a member of the ACM.
E-mail: cleary@eecis.udel.edu



Michael Paterakis was born in Athens, Greece, in 1961. He received his Diploma degree from the National Technical University of Athens, Greece, his M.Sc. degree from the University of Connecticut, and his Ph.D. degree from the University of Virginia, in 1984, 1986, and 1988, respectively, all in electrical engineering. Since 1994, he is an Associate Professor of Telecommunications and Computer Networks in the Department of Electronic and Computer Engineering at the Technical University of Crete, Greece. Before that, he was an Associate Professor in the Department of Computer and Information Sciences at the University of Delaware, USA, where he was since September 1988. His research interests include computer communication networks with emphasis on protocol design, modeling and performance evaluation of broadband high speed multimedia networks, of multiple access wireless microcellular communication systems, and of packet radio networks; queueing and applied probability theory and their application to computer systems and communication networks. He has published fifty research articles in archival journals and refereed conference proceedings in the above mentioned technical areas. He served on the Technical Program Committees of the 1991 International Conference on Distributed Computing Systems, Arlington, Texas, the 1992 IEEE INFOCOM Conference, Florence, Italy, and the 1994 IEEE INFOCOM Conference, Toronto, Canada. Professor Paterakis is a senior member of the IEEE. He is also a member of the IEEE Technical Committee on Computer Communications, the Greek Chamber of Professional Engineers, and the Greek Association of Electrical and Electronic Engineers.
E-mail: pateraki@telecom.tuc.gr