International Journal of Wireless Information Networks, Vol. 5, No. 1, 1998

Voice/Data Integrated Wireless Channel Access in Third Generation Digital Cellular Networks: The Performance of Bursty Data Generated by Interactive Applications

I. Rombogiannakis[1] and M. Paterakis[1]

Mobile wireless communications, which include cellular telephones, land mobile radio, and personal communications systems, have experienced enormous growth over the last decade. Data services represent a critical component of future wireless communications, but have received little attention so far. While some attention has been given to specialized mobile data networks, less has been directed at the ongoing design of data services in evolving third generation digital cellular wireless networks. In this work we present the results of a simulation study that explores the performance of a Reservation Random Access (RRA) scheme for transmitting data packets over a common radio broadcast channel in a cellular radio environment. In addition to voice traffic, we consider data packet traffic generated by interactive applications. It is expected that such applications will be very important in third generation wireless access mobile communication systems. Through an extensive simulation study of the data message delay distribution, we show that the proposed RRA scheme, originally designed under the Poisson data message arrival process assumption, can also efficiently operate and be optimized under the extremely bursty data message arrival process characterized by independent, identically distributed, Pareto message interarrival times.

KEY WORDS: Wireless cellular networks; integrated channel access; reservation random access schemes; Pareto distribution; simulation-based performance study.

1. INTRODUCTION

The field of wireless communications is expanding rapidly as a result of advances in digital communication systems, cellular networks, portable computers, and personal communications systems. Current wireless networks are optimized for voice communications. In recent years rapid advances in microelectronics have led to a proliferation of versatile computers and communication equipment. This rapid proliferation, especially in the area of personal and portable computers, creates an enormous demand for data communication. It is expected that in the near future, data services and applications such as electronic mail, file transfer, and database query will require access over wireless cellular networks.

The radio channel bandwidth within today's cellular systems is limited, demand for access continues to surge, and presently no substantial bandwidth increase is predicted. System capacity can be increased by using microcells to increase frequency reuse and by using efficient multiple-access protocols to exploit the variations in access and service required by disparate sources.

In densely populated areas, microcells are expected to end the distinction between cordless and mobile telephones and to provide access to broadband ISDN public networks for large numbers of mobile voice and data terminals. Mobile communication within a microcell entails slow-moving vehicles (i.e., vehicles in city traffic) and/or pedestrians equipped with inexpensive,

[1] Electronic and Computer Engineering Department, Technical University of Crete, 731-00 Chania, Greece; e-mail: {rombo, pateraki}@telecom.tuc.gr.

1068-9605/98/0100-0001$15.00/0

lightweight, and low-powered communication devices (e.g., telephones, palmtop PCs, etc.). Microcell diameters are predicted to be on the order of 100 m, and mobile wireless networks require call setup and handoff capabilities. Therefore, using microcells will increase the processing requirements of network control. It is expected that call control and call management functions will be decentralized and highly distributed [1].

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 channel feedback information, and serves as an interface to the mobile switching center. Since base-to-mobile transmissions can be supported by a time-division-multiplexed (TDM) broadcast channel, we focus on the mobile-to-base channel transmissions.

Speech alternates between periods of talk (called talkspurts) and silence; and voice packet transmissions are subject to a real-time constraint. We assume that each voice terminal uses voice activity detection [2] to suppress its silence transmissions. Thus, voice terminals only require channel access during talkspurt. Voice packet delay requirements are stricter than those for data packets. This is because delays in voice are very annoying to a listener. Each voice packet must be delivered within a specified maximum delay (typically in the range 150–200 ms) to be useful. If the delay of a voice packet is greater than the maximum specified delay, then the voice packet is dropped. Speech can withstand a small (1–2%) amount of dropped packets without suffering large quality degradation [3], at least one which can be perceived by humans. Data traffic, on the other hand, is often extremely bursty and more tolerant of delays (e.g., delays up to 200 ms are often acceptable), but 100% delivery of correct packets is often required (e.g., in the case of a file transfer).

In this paper we consider interactive data applications. In such scenarios a typical data user might do things like read/compose short e-mail messages, respond to paging-type messages, transfer files between the mobile and a fixed computer, and query some form of database. The analytic traffic model used in our study is taken from refs. 4–7 and it fits measured Ethernet LAN network traffic [7], wide-area TCP live network traces of interactive Telnet applications [4], and wide-area network traffic measurements of applications such as WWW and Multimedia Backbone (Mbone) [6]. Although the operation environments for wide-area and local area networks may be different from the context of wireless mobile communication systems, the model used is important since it describes the user behavior associated with interactive data applications [5]. Such applications will gain in importance in the wireless communications arena as small, portable, and inexpensive computing devices proliferate.

The above-mentioned empirical studies of traffic measurements from a variety of packet networks have convincingly demonstrated that actual network traffic is self-similar in nature (i.e., bursty over a wide range of time scales). Self-similar traffic exhibits correlations over a wide range of time scales and therefore it has *long-range dependence*, while traditional traffic models (e.g., Poisson message arrivals) are *short-range dependent* in nature (i.e., they typically focus on a very limited range of time scale). In refs. 7 and 8 the authors show that the superposition of many alternating independent and identically distributed ON/OFF sources whose ON and OFF periods have high variability or infinite variance results in self-similar aggregate traffic. As the mathematical vehicle for modeling such phenomena they used *heavy-tailed* distributions with infinite variance (e.g., the Pareto distribution).

Therefore, in this paper the message interarrival times are assumed independent, identically distributed according to a *Pareto* distribution. The Pareto distribution is a distribution with memory, heavy tail, and strong burstiness [9]. Depending on the value of one of its parameters, it can have finite mean and infinite variance. The strong burstiness of such a message arrival process leads to severe channel multiple-access operational conditions capable of providing us with indications on the performance robustness of the random multiple-access algorithms we study to the characteristics of the input message arrival process.

The organization of the paper is as follows. In Section 2 we introduce the system, data traffic, and voice traffic models, and the voice and data packet transmission protocols. The simulated system parameters, the optimal choice of an important algorithmic parameter, and representative simulation results are presented and discussed in Section 3. The paper is concluded in Section 4.

2. SYSTEM MODEL

The channel time 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. Each frame, as illustrated in Fig. 1, consists of three intervals: one voice request inter-

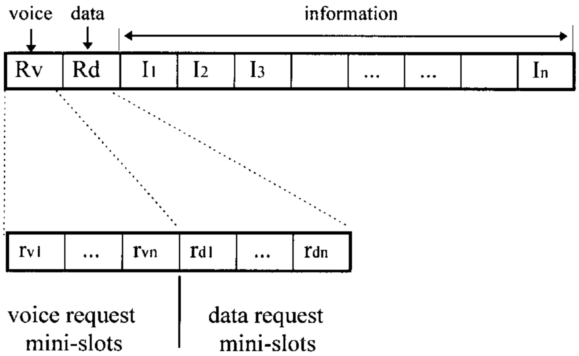

Fig. 1. The voice, data and information intervals within a frame.

val, one data request interval, and an information interval. Within the information interval, each slot accommodates exactly one fixed-length packet that contains voice (or data) information and a header. Both of the request intervals are subdivided into minislots; and each minislot accommodates exactly one fixed-length request packet. For both voice and data traffic, the request packet must include a source identifier. For data traffic, the request packet might also include message length and quality of service parameters such as priority and required slots/frame. We simplify our presentation by assuming that both request intervals contain an equal number of minislots, and that data terminals need only 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 minislot that indicates only the presence or absence of a collision within the minislot (collision (C) versus no collision (NC)). Since the feedback packet is small (several bits) and the microcell diameter is on the order of 100 m, the feedback information can be assumed available to terminals before the next minislot. Upon successfully transmitting a request packet, the terminal waits until the end of the frame to learn of its reservation slot. If the terminal is not successful within the request interval, it attempts to transmit again in the request interval of the next frame. A terminal with a reservation transmits freely during its reserved slot.

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 the quality of service parameters. Upon a 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 (handoff), the base station will delete the table entry after some prescribed period of time.

We focus on steady-state channel access and assume the following about the allocation and queuing policies. When servicing a request, the base station always allocates the earliest empty information slot within the frame. Next, we assume that voice has a higher priority than the data traffic (this is reasonable for near future). Thus, the base station will service all of the outstanding voice requests before servicing any data request. Within each priority class, the queuing discipline is assumed to be First Come, First Served (FCFS). Finally, if any new voice requests are received and the frame is full, 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 a data request at the front of the data request queue.

2.1. Data Traffic Characterization

While there are satisfactory models to represent the packet arrival process of a voice source (i.e., to model the behavior of a speech activity detector), for a variety of interactive data applications (e.g., file transfer, e-mail, and database query) an equally satisfactory model is not easily available. Furthermore, a number of studies have shown that in a LAN/WAN environment the packet interarrival times of the aggregate traffic clearly differ from exponential [4, 6, 7].

Following ref. 5, data message interarrival times are assumed Pareto distributed with parameters a, k:

$$P\{T \le t\} = 1 - \left[\frac{k}{t}\right]^a, \qquad k, a \ge 0, \quad t \ge k$$

The parameters of the Pareto distribution are the *location parameter* k, which represents the minimum time between the arrivals of two consecutive messages, and the *shape parameter* a. It can be easily shown that if $a \leq 2$, the distribution has infinite variance, while if $a \leq 1$, the mean is infinite as well [9]. Therefore, the distribution is heavy tailed, with infinite variance and mean (a distribution is called heavy tailed if the ratio $P\{T \geq t\}/t^{-a}$ tends to 1 as $t \to \infty$, $a \geq 0$). A more general definition of heavy-tailed distributions defines a distribution as a heavy-tailed if the mean conditional exceedance of the random variable T

$$E[T - t \mid T \geq t]$$

is an increasing function of t [9]. For a Pareto-distributed random variable T with $a > 1$ (i.e., with finite mean), the mean conditional exceedance time is a linear function of t [9, p. 70]:

$$E[T - t \mid T \geq t] = \frac{t}{a - 1}$$

Using this second definition, consider a Pareto-distributed random variable, and assume that it represents the waiting time of a customer in a service facility. Then the longer the customer has waited in the service system, the longer is its expected additional waiting time. Contrast this behavior with waiting times distributed according to a light-tailed distribution (such as the uniform, for which the mean conditional exceedance is a decreasing function of t), or to a medium-tailed distribution (such as the memoryless exponential, for which the mean conditional exceedance is independent of t, the waiting time so far) [9].

These above-mentioned mathematical properties account for the strong burstiness of the packet arrival process characterized by independent Pareto-distributed packet (message) interarrival times. This in turn leads to severe channel multiple-access operational conditions, capable of providing us with indications on the performance robustness of the random multiple-access algorithm we study to the characteristics of the input packet arrival process.

2.2. Transmission Protocols

The algorithms below were selected to investigate performance aspects of the proposed RRA scheme, rather than optimization for a set of system parameters.

2.2.1. Voice Terminals

Voice terminals use the *ideal* algorithm for transmitting their request packets. According to the ideal algorithm, all voice request packets present at the start of the voice request interval are correctly received by the base station by the end of the voice request interval. We examine this protocol 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 beginning of a talkspurt and the end of the first voice packet transmission of this talkspurt into a reserved slot), over all the implementable voice transmission protocols.

2.2.2. Data Terminals

To transmit data request packets, 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 [10]. Its details are given later in this section. Throughout this paper, we will use the term data 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.

The algorithm uses a blocked access mechanism established by the following first-time transmission rule for newly generated data messages. Terminals with new data message arrivals may not transmit during a collision resolution period (CRP). A collision resolution period 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 minislot). In the first minislot following a CRP, all of the terminals whose data message arrived within a prescribed allocation interval of maximum length Δ transmit their request packets. During some CRP which starts with a collision minislot, each terminal which is involved in that collision uses a counter r as follows:

1. Contenders with $r = 0$ transmit into the request minislot.
2. Let x be the feedback for that transmission. Then:
   a. If x = no collision:

if $r = 0$, the request packet was transmitted successfully;
if $r = 1$, then $r = 0$.

b. If x = collision:
if $r = 0$, then reinitialize r to 0 or 1 with probability 0.5[2];
if $r = 1$, then $r = 1$.

3. Repeat steps 1–2 until two consecutive feedbacks indicating noncollision occur (end of CRP).

The operation of this protocol can be depicted by a two-cell stack, where in a given request slot the bottom cell contains the transmitting terminals (those with $r = 0$) and the top cell contains the withholding terminals (those with $r = 1$). Here, we must emphasize that the stack is a creation of our imagination and helps us to visualize the operation of the algorithm, it is not an entity maintained by any terminal in the system or the base station.

An example of the operation of the algorithm is given in Fig. 2, where there are three data contending terminals which try to transmit their data request packets at the beginning of a frame.

All terminals in the bottom cell (those with $r = 0$) transmit their request packets in the first minislot, and a collision ensues. The subsequent transitions in time of the stack (based on counter reinitialization for the contending terminals chosen for illustrative purposes) follow the rules 1–3. Notice that two consecutive no-collision (NC) feedbacks indicate an empty stack, and therefore the end of the collision resolution period (CRP) which started with the initial collision among the three request packets.

[2]The splitting probability for the contending terminals in step 2b of the algorithm can be optimally chosen to maximize the algorithmic throughput. In ref. 19 it is reported that the optimal value of the splitting probability is $p^* = 0.544$ and the corresponding maximum throughput is equal to 0.43067 (a 2% increase over the throughput of the algorithm when the splitting probability p is taken to be equal to 0.5).

2.3. Voice Traffic Model

The primary voice traffic model assumptions are as follows.

1. Voice terminals are equipped with a slow voice activity detector [2, 11] 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 all of the voice transitions (e.g., talk/silence) occur at the frame boundaries. Therefore, the speech activity is modeled with a two-state discrete-time Markov chain as shown in Fig. 3. The talkspurt-to-silence transition probability is denoted by p_{TS}, and the silence-to-talkspurt transition probability is denoted by p_{ST}. The talkspurt and silence periods are geometrically distributed with mean $1/p_{TS}$ and $1/p_{ST}$ 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:

$$p_T = p_{ST}/(p_{ST} + p_{TS}), \qquad p_S = 1 - p_T$$

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 [12].
3. The voice delay limit is equal to the duration of

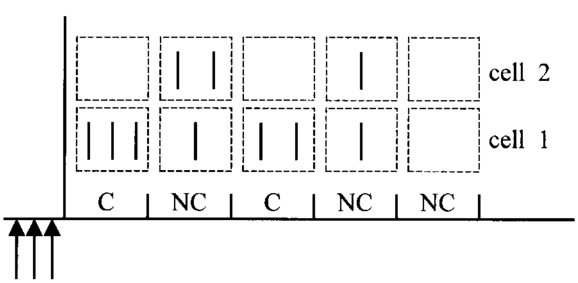

Fig. 2. Two-Cell Stack algorithm; an example of its operation.

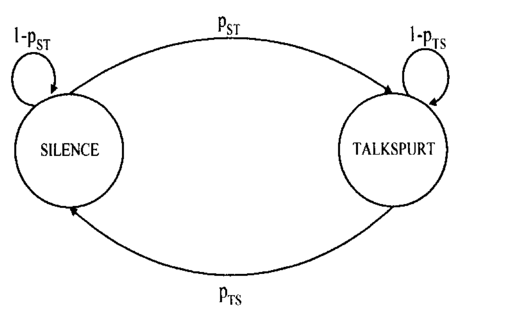

Fig. 3. The voice traffic model.

two time frames. Assumptions 1 and 3, 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 transmit successfully a request packet during the voice request interval will drop one voice packet.

4. The channel is error-free and without capture. Additionally, the base station correctly broadcasts to the terminals the pending resource allocations before the beginning of the next frame. As a result, errors within the system only occur when two or more request packets are transmitted simultaneously (collision) within the same request minislot.
5. Reserved slots are deallocated immediately. This implies that a terminal holding a reservation signals the base station upon the completion of a talkspurt (or data message). Delayed deallocation schemes can be evaluated with minor modifications to the simulator presented in this paper. In a delayed deallocation scheme, the base station determines the end of a talkspurt (or data message) by detecting silence in a reserved slot (i.e., one information slot is wasted per voice talkspurt or data message).

2.4. System State Transitions

As shown in Fig. 4, we will describe an active voice terminal as being in one of four states: silent, contender, queued, or reserved. A silent terminal has no packet to transmit and it does not require channel resources. On transition into talkspurt, the terminal enters the contender state and remains there until it successfully transmits a request packet. The corresponding time interval is equal to an integer number of frames; consider j frames, $j \geq 1$, during which the voice terminal drops $j-1$ packets from its talkspurt. In case of the ideal algorithm for voice request packet transmissions, $j = 1$, and no voice packet is dropped. Since the requests are queued at the base station, the terminal enters the queued state and remains there until the base station provides it with a reservation (as a minimum, this will be the end of the frame in which the request was received). After receiving a reservation, the terminal enters the reserved state and transmits one voice packet per frame into its allocated slot until it exhausts its packets and returns to the silent state.

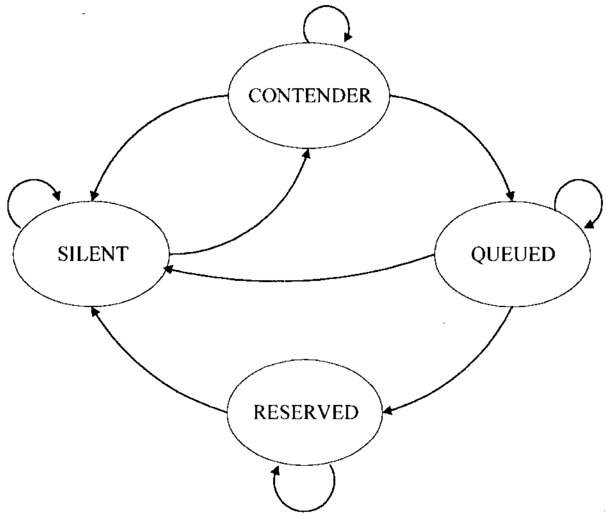

Fig. 4. The state transition diagram for an active voice terminal.

3. SIMULATIONS

3.1. System Parameters

We use the parameters contained in Table I to evaluate the system performance via simulations. The channel rate is from refs. 13 and 14. The speech rate value assumes the use of a 32 kb/s adaptive differential pulse code modulation (ADPCM) in the microcellular environment [15]. The packet size (53 bytes) was selected for compatibility with ATM networks [17]. The values for the mean talkspurt/silence durations produce a speech activity of approximately 44% [11]. The frame duration is 12 ms, and it accommodates approximately 51 slots (i.e., information slots). The voice delay limit is 24 ms, equal to the duration of two frames. To account for guard time and synchronization, we assume that 50 slots are available per frame. The first and second of these slots are dedicated to voice and data request intervals, respec-

Table I. Experimental System Parameters

Speech rate	32 kb/s
Channel rate	1.2 Mb/s
Packet size (bits)	424
Speech/data (bits)	384
Header (bits)	40
Frame duration (bits)	12
Slots per frame	50
Voice delay limit (ms)	24
Mean talkspurt duration (s)	1.41
Mean silence duration (s)	1.78

tively, and they are both subdivided into minislots. The remaining 48 slots comprise the information interval of the frame. We chose six minislots per request interval (so that each minislot 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 per frame, etc.).

Table II. The Mean Packet Delay (in Slots) and the Length of the Unexamined Interval (in Slots) at the End of the Simulation, when the Location Parameter of the Pareto Distribution is $k = 0.7$

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	λ	$E(D)$	lag	$E(D)$	lag	$E(D)$	lag
1.2	0.238	12.12	1	10.51	1	10.88	16
1.3	0.329	40.71	43	31.47	1	32.9	22
1.34	0.362	101	76	69.68	5	71.77	39
1.36	0.378	187.2	281	131	25	155	94
1.37	0.385	316.2	420	172	53	269	209
1.38	0.393	570.6	706	419	319	367	222
1.39	0.40	1176	6364	534	331	2039	450
1.4	0.408	5796	12933	2122	1180	6463	7835

3.2. The Choice of the Parameter Δ of the Two-Cell Stack Algorithm

Following ref. 10, for Poisson packet arrivals, the optimal value of Δ which maximizes the throughput of the Two-Cell Stack algorithm for data request packets transmission is 2.33 slots, and the respective throughput is 0.429 packets per slot. Recall that Δ corresponds to the maximum size of the allocation interval in the first minislot following a CRP.

The choice of optimal value of Δ for Pareto-distributed message interarrival times has not been examined before. In this paper, we chose simulation as the means for determining the optimal value of Δ in the case where message interarrival times are Pareto distributed.

Simulation was executed on a desktop UNIX workstation (SUN Sparcstation 5). Each run simulates the successful transmission of $Z = 1$ million packets. Z was chosen so that the examined algorithm reaches steady state. In many cases, we have simulated smaller and larger Z values (e.g., 500,000 and 2,000,000 packets, respectively). In all of these cases, we observed that by simulating 1,000,000 successfully transmitted packets we were able to estimate satisfactorily the steady-state algorithmic performance metrics.

In Tables II–IV we present the simulation results of the operation of the Two-Cell Stack algorithm for various values of the window size Δ. It is worthwhile noticing that we are interested in the value of Δ which optimizes the operation of the Two-Cell Stack algorithm for a set of different values of the location parameter k of the Pareto distribution which are of interest to us ($k \geq 0.6$). Table II contains the simulation results for the mean packet delay $E(D)$ and the length of the unexamined interval (lag) at the end of the simulation (i.e., when Z packets have been successfully transmitted), when the location parameter of the Pareto distribution is equal to $k = 0.7$. From Table II, we observe that when $\Delta = 1.6$ slots, the algorithm produces the smallest values for mean packet delay and unexamined interval. As we will see from the simulation results in Tables III and IV, the value of $\Delta = 1.6$ does not absolutely correspond to the optimum; however, it corresponds to the central point of a neighborhood of values which optimize the performance of the Two-Cell stack algorithm for all values of the parameter k which are of interest to us.

Table III contains the simulation results when the

Table III. The Mean Packet Delay (in Slots) and the Length of the Unexamined Interval (in Slots) at the End of the Simulation, when the Location Parameter of the Pareto Distribution is $k = 0.6$

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	λ	$E(D)$	lag	$E(D)$	lag	$E(D)$	lag
1.05	0.079	16.97	14	16.43	9	18.89	30
1.1	0.151	30.48	100	28.8	37	35.58	55
1.15	0.217	63	129	61.52	150	93.15	106.6
1.17	0.242	99.89	222	94.80	218	166.83	130
1.2	0.277	225.13	404.5	259.33	624	636.74	1833
1.22	0.30	591.52	238.5	559.5	325	1942	4424
1.25	0.333	2804	4456	2402	5696	35304	67636

location parameter of the Pareto distribution is $k = 0.6$. It is easily seen that the optimal value of the parameter Δ remains around 1.6 slots; however, the mean packet delay is almost the same with that for $\Delta = 1.8$ slots. This fact could have led us to choose the value of Δ to be 1.7 or 1.8 slots, if what we wanted was the optimal value of Δ when k only takes the value $k = 0.6$.

We can draw very similar conclusions from the simulation results for $k = 0.8$. From Table IV we observe that the optimal Δ value is 1.4 slots. Notice, however, that the corresponding mean packet delay values for $\Delta = 1.6$ and 1.4 slots are very close to each other only when the packet arrival rate does not exceed 0.433 packets per frame.

From the above discussion it is clear that the optimal Δ value under the earlier stated objective (i.e., that we want to choose a single value of Δ which optimizes the operation of the Two-Cell Stack algorithm for a set of values of the location parameter k of the Pareto distribution which are of interest to us) is 1.6 slots.

3.3. Voice-Data Integration

Simulation was chosen as the means for evaluating the performance of the data traffic. The fact that the Pareto-distributed message interarrival time is a random variable with memory greatly complicates the analysis of the system compared to the corresponding analysis when the data message arrival process is Poisson [16]. Simulations were executed on a desktop UNIX workstation (SUN Sparcstation 5).

All simulations consist of ten independent runs of $L = 305{,}000$ frames each. To reduce startup effects, the first 5000 frames serve as the warmup period. L was chosen so that the examined system reaches steady state. In many cases, we simulated smaller and larger L values, as well as a larger number of independent runs. In all of these cases we observed that by simulating ten independent runs of $L = 305{,}000$ frames each, we were able to estimate satisfactorily the steady-state data packet delay performance metrics of interest. During each run the number of terminals within the system is constant, the terminals are initially silent, and the results for specific performance measures are calculated with cumulative data.

We assume that data messages are generated by a large unknown number of data terminals (theoretically infinite) and that (a) the aggregate message arrivals are Poisson distributed with rate λ messages/frame, or (b) the message interarrival times are independent, identically distributed Pareto random variables. Additionally, we assume that the message length is geometrically distributed with parameter q (mean $B = 1/q$). In the simulations we set $q = 1/8$ ($B = 8$). Here, we would like to note that very similar results were also observed with deterministic message size equal to eight packets [18].

From the simulation results of the voice traffic in [16] with Poisson-distributed data message arrival times, we observe that when the number of active voice terminals is smaller than or equal to 86 terminals ($N \leq 86$) the voice dropping probability (the ratio of the average number of voice packets dropped per frame to the average number of voice packets generated per frame) is almost negligible. When $N > 87$ the voice packet dropping probability starts to increase rapidly, and for $N > 97$ is bigger than 1%. For that reason, in our presentation we will consider N values in the range $86 \leq N < 97$.

In Fig. 5, we provide the simulation results for the mean data wait message delay parametrized on the location parameter of the Pareto distribution k, as a function

Table IV. The Mean Packet Delay (in Slots) and the Length of the Unexamined Interval (in Slots) at the End of the Simulation, when the Location Parameter of the Pareto Distribution is $k = 0.8$

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	λ	$E(D)$	lag	$E(D)$	lag	$E(D)$	lag
1.4	0.357	7.76	1	5.38	1	5.24	1
1.5	0.416	23	15.54	10.4	1	9.4	2.5
1.55	0.433	76.6	96.8	19.65	8.5	17.1	10
1.6	0.468	2639	966	45.44	25.5	74.73	12
1.64	0.487	30067	57350	301	141.5	106.9	169
1.645	0.490	—	—	386	200	132.67	190
1.65	0.492	—	—	749	249	336	453
1.66	0.496	—	—	4196	5678	1597	679

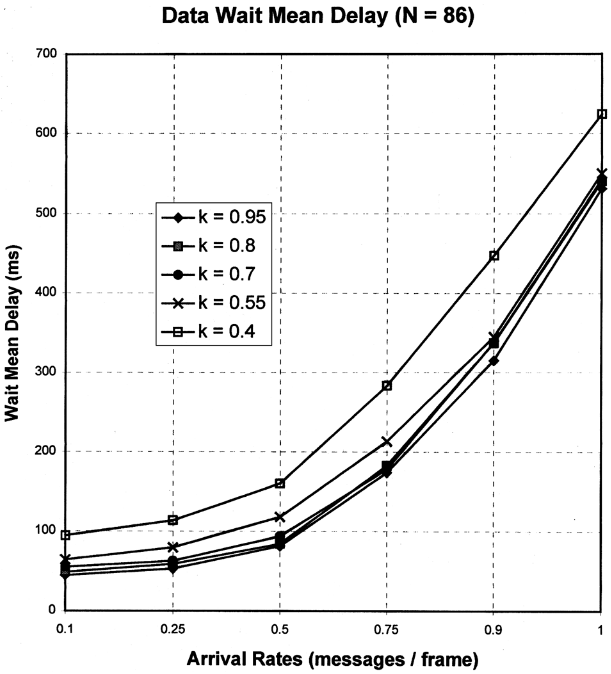

Fig. 5. The mean data message wait delay parametrized on the location parameter of the Pareto distribution k, as a function of the arrival rate λ, for a system with $N = 86$.

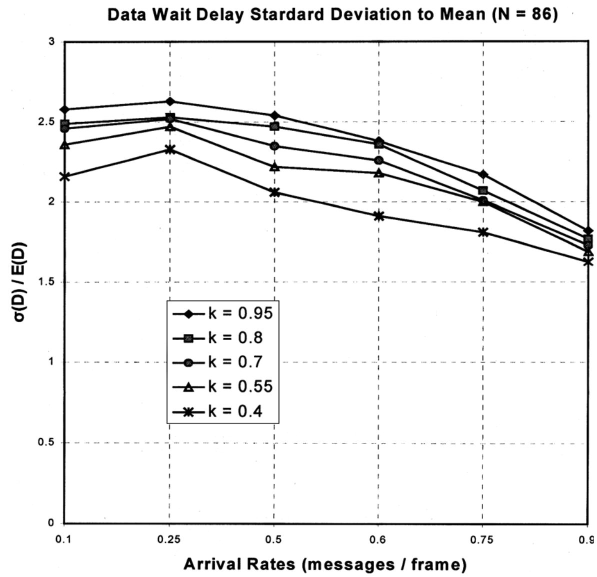

Fig. 6. The ratio of the standard deviation to the mean of the data message wait delay as a function of the data message arrival rate λ, for different values of the location parameter k of the Pareto distribution.

of the arrival rate λ, for a system with $N = 86$ active voice terminals. The data wait message delay is defined as the time interval from the arrival instant of a data message until the end of the successful transmission of the first packet of the same message (i.e., until that message has reserved an information slot within the frame to transmit its packets). We observe that for all values of k (except $k = 0.4$, which is a small value of k for which two message arrivals per frame are very possible), the mean data wait delay remains below 200 ms for arrival rates less than 0.75 messages/frame. Furthermore, we observe that for a given data message arrival rate, the mean data wait delay is an increasing function of the location parameter k of the Pareto distribution.

In Fig. 6 we present the simulation results for the ratio of the standard deviation to the mean of the data message wait delay parametrized on the location parameter k of the Pareto distribution, versus the message arrival rate λ. We observe that the ratio decreases with increasing λ, except when λ is small (e.g., $0.10 \leq \lambda \leq 0.25$), where the ratio slightly increases. The latter increase is due to the fact that both the mean and the standard deviation of the message wait delay are increasing functions of λ, and in the interval $\lambda \in [0.1, 0.25]$ the standard deviation of the message wait delay increases faster than the mean (while the reverse is true for $\lambda > 0.25$) [18].

Figure 7 shows the data message wait delay distribution parametrized on the location parameter k of the Pareto distribution for a given data message arrival rate $\lambda = 0.25$ messages/frame and for $N = 86$ active voice terminals. The notation $[x–y]$ on the horizontal axis denotes the time interval from x to y frames. We observe that for all values of the parameter k, a large fraction of data messages experience wait delays below 3 frames (or 36 ms). For example, for $k = 0.8$, approximately 85% of messages experience delays below 36 ms. In addition in this case, 90% of the messages experience delays below 192 ms (16 frames long). In addition, we observe that as k decreases, the data packet wait probability mass shifts to larger values. Figure 8 shows the message wait delay distribution for a system with $N = 86$ and 90, respectively, and for a data message arrival rate $\lambda = 0.50$ messages/frame. From this figure we can see that as the number of active voice terminals increases, the number of information slots available to the data decreases, hence the data wait probability mass shifts to larger delay values.

In Fig. 9 we present the simulation results for the mean data message wait delay versus the data message arrival rate λ for three distributions, Pareto 1, Pareto

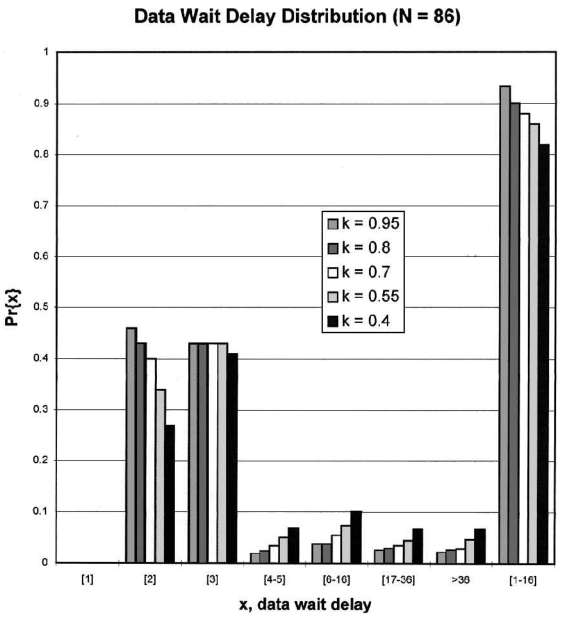

Fig. 7. The data message wait delay distribution parametrized on the location parameter of the Pareto distribution k for a data arrival rate $\lambda = 0.25$ messages/frame, when $N = 86$.

2, and Poisson, and for N = 86. Pareto 1 is defined to be the Pareto distribution with location parameter k = 0.95, while Pareto 2 is defined to be the Pareto distri-

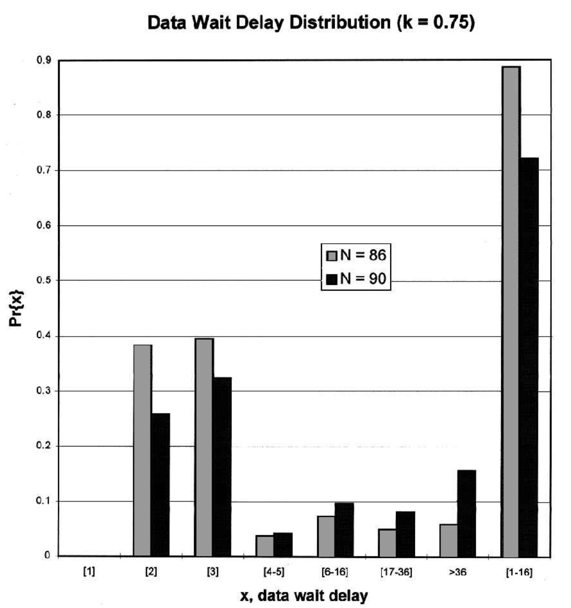

Fig. 8. The data message wait delay distribution for a data arrival rate $\lambda = 0.50$ messages/frame when $N = 86$ and 90.

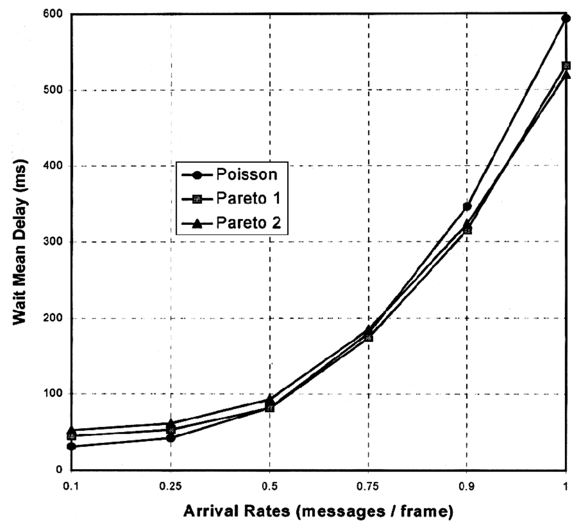

Fig. 9. The data message mean wait delay as a function of the data message arrival rate λ for the distributions Pareto 1, Pareto 2, and Poisson when $N = 86$.

bution with location parameter k = 0.75. We observe that for arrival rates less than 0.50 messages/frame the mean data message wait delay is smallest for Poisson message arrivals, while for arrival rates larger than 0.75 messages/frame, the Pareto 1 and Pareto 2 cases incur the smallest mean data message wait delays. Furthermore, we observe that the mean data message wait delay is maintained below 200 ms for message arrival rates below 0.75 messages/frame in all three of the examined distributions.

Figure 10 shows the data message wait delay distribution for the three distributions (Pareto 1, Pareto 2, Poison) when N = 86 and for an arrival rate equal to 0.50 messages/frame. We observe that a significant fraction of data messages (90% for the Poisson, and 89% for the Pareto 1 distribution) experience wait delays below 3 frames (or 36 ms), and a very big fraction of data messages (97% for the Poisson and 93% for the Pareto 1 distribution) experience delays below 200 ms.

Finally, in Fig. 11 we present the simulation results for the ratio of the standard deviation to the mean data message wait delay as a function of the data message arrival rate λ for the Pareto 1, Pareto 2, and Poisson distributions when $N = 86$. We observe that this ratio varies between 2.65 and 1.62 for $0.10 \leq \lambda \leq 0.9$. Furthermore, as λ increases, the ratio decreases (except for the Pareto case with large k and small λ values, e.g., the Pareto 1 with $0.10 \leq \lambda \leq 0.25$, where the ratio slightly increases).

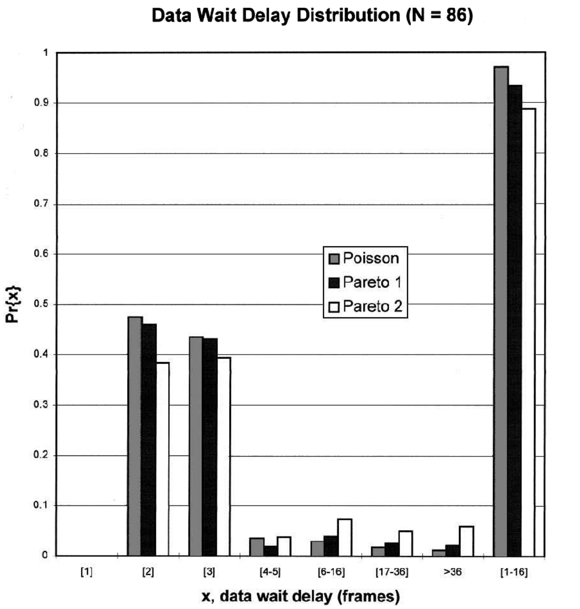

Fig. 10. The data message wait delay distribution for the distributions Pareto 1, Pareto 2, and Poisson when $N = 86$ and for $\lambda = 0.50$ messages/frame.

These results imply that the wait delay of a randomly selected message may be as high as 2.65 times the mean data wait delay. The latter observation led us to the conclusion that knowledge of the data message mean wait delay alone does not suffice to predict accurately the wait delay of a randomly selected data message.

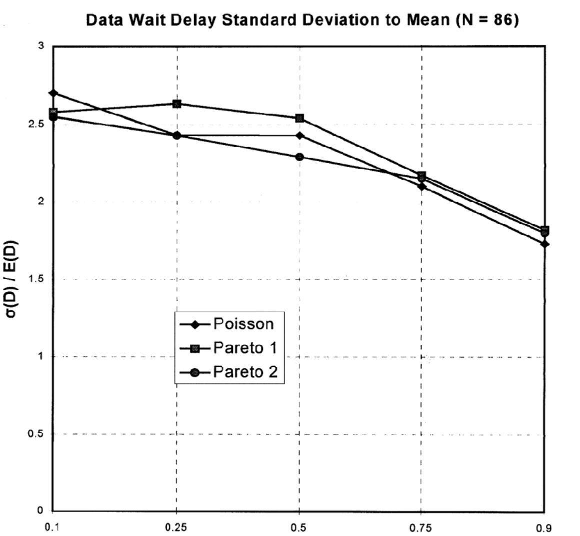

Fig. 11. The ratio of the standard deviation to the mean of the data message wait delay as a function of message arrival rate λ for the distributions Pareto 1, Pareto 2, Poisson.

4. CONCLUSIONS

In this work we have evaluated the performance of an RRA scheme for integrating voice and data traffic channel access in outdoor microcellular environments. The scheme was originally proposed in ref. 16. The problem considered in this paper is interesting because the data traffic model we use (with Pareto-distributed message interarrival times) has been found to fit well wide-area network interactive bursty data applications. Such applications are expected to be a critical component of third generation digital wireless cellular networks.

Through an extensive simulation study we have shown that RRA schemes originally designed and evaluated under the Poisson data message arrival process assumption can also efficiently operate and be optimized under the extremely bursty packet arrival process characterized by Pareto message interarrival times.

REFERENCES

1. D. J. Goodman, Cellular packet communications, *IEEE Transactions on Communications*, Vol. COM-38, No. 8, pp. 1272–1280, 1990.
2. S. Nanda, D. J. Goodman, and U. Timor, Performance of PRMA: A packet voice protocol for cellular systems, *IEEE Transactions on Vehicular Technology*, Vol. 40, pp. 584–598, 1991.
3. B. Mukherjee, Integrated voice-data communication over high-speed fiber optic networks, *IEEE Computer Magazine*, Vol. 24, No. 2, pp. 49–58, 1991.
4. V. Paxson and S. Floyd, Wide area traffic: The failure of Poisson modeling, *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, pp. 226–244, 1995.
5. G. Anastassi, D. Grillo, L. Lenzini, and E. Mingozzi, A bandwidth reservation protocol for speech/data integration in TDMA-based advanced mobile systems, *International Journal of Wireless Information Networks*, Vol. 3, No. 4, pp. 243–252, 1996.
6. W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 1, pp. 71–86, 1997.
7. W. Lerand, M. Taqqu, W. Willinger, and D. Wilson, On the self-similar nature of the Ethernet traffic (extended version), *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, pp. 1–16, 1994.
8. M. Taqqu, W. Willinger, R. Sherman, and D. Wilson, Proof of a fundamental result in self-similar traffic modelling, *Computer Communications Review*, Vol. 27, pp. 5–23, 1997.
9. B. Arnold, *Pareto Distributions*, International Cooperative Publishing House, Baltimore, MD, 1983.
10. M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, A full sensing window RA algorithm for networks with strict delay constraints, *Algorithmica* , Vol. 4, pp. 321–328, 1989.

11. P. T. Brady, A technique for investigating on–off patterns of speech, *Bell Systems Technical Journal*, January 1965.
12. T. Suda and T. Bradley, Packetized voice/data integrated transmission on a token passing ring local area network, *IEEE Transactions on Communications*, Vol. COM-37, No. 3, pp. 238–244, 1989.
13. J. M. DeVile, A reservation multiple access scheme for a future universal mobile telecommunications system. In *Proceedings of the 7th IEE Conference on Mobile and Personal Communications, Brighton, UK*, pp. 210–215, December 1993.
14. J. Dunlop, J. Irvine, D. Robertson, and P. Cosimini, Performance of a statistical multiplexed access mechanism for a TDMA radio interface, *IEEE Personal Communications*, Vol. 2, No. 3, pp. 56–64, 1995.
15. D. C. Cox, Wireless personal communications: What is it?” *IEEE Personal Communications*, Vol. 2, No. 2, pp. 20–35, 1995.
16. A. C. Cleary and M. Paterakis, Design and performance evaluation of an RRA scheme for voice data channel access in outdoor microcellular wireless environments. In B. Jabbari, P. Godlewski, and X. Langrange (eds.), *Multiaccess, Mobility and Teletraffic for Personal Communication*, Kluwer, Dordrecht, pp. 117–130, 1996; an extended version is to appear in the *ACM Wireless Networks Journal*.
17. Raif Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, Boston, 1994.
18. I. Rombogiannakis, Design and performance evaluation of an RRA protocol for voice-data integration in microcellular wireless environments, Diploma Thesis, Department of Electronics and Computer Engineering, Technical University of Crete, Greece, 1996 [in Greek].
19. M. Paterakis and P. Papantoni-Kazakos, A simple window random access algorithm with advantageous properties, *IEEE Transactions on Information Theory*, Vol. 35, No. 5, pp. 1124–1130, 1989.

Ioannis Rombogiannakis was born in Heraklion, Crete, Greece, in 1973. He received his Diploma degree in electronics and computer engineering from the Technical University of Crete, Greece, in 1996, where he is currently working toward his M.Sc. degree. His research interests focus on the design and performance evaluation of computer communication and multimedia information systems, and queueing and applied probability theory and their application to computer communication and information systems.

Michael Paterakis 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 has been an Associate Professor in the Department of Electronic and Computer Engineering at the Technical University of Crete, Greece. Before 1994, he was an Associate Professor in the Department of Computer and Information Sciences at the University of Delaware, where he began in September 1988. During the summer of 1991, he was a Visiting Professor in the Department of Electronic and Computer Engineering, Technical University of Crete, Greece.

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, as well as queueing and applied probability theory and their application to computer communication networks. He has published extensively in archival journals, refereed conference proceedings, and edited books 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, the 1994 IEEE INFOCOM Conference, Toronto, Canada, and the 1997 4th IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'97), Sani Beach, Chalkidiki, Greece.

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.