Paging and Registration in Cellular Networks: Jointly Optimal Policies and an Iterative Algorithm

Bruce Hajek
Department of Electrical and
Computer Engineering
University of Illinois at Urbana Champaign
1308 W. Main Street, Urbana, IL 61801
Email: b-hajek@uiuc.edu

Kevin Mitzel
Sirius Satellite Radio
Milford, MI 48381
Email: kmitzel@siriusradio.com

Sichao Yang
Department of Electrical and
Computer Engineering
University of Illinois at Urbana Champaign
1308 W. Main Street, Urbana, IL 61801
Email: syang8@uiuc.edu

***Abstract*— This paper explores optimization of paging and registration policies in cellular networks. Motion is modeled as a discrete-time Markov process, and minimization of the discounted, infinite-horizon average cost is addressed. The structure of jointly optimal paging and registration policies is investigated through the use of dynamic programming for partially observed processes. It is shown that there exist policies with a certain simple structure that are jointly optimal, though the dynamic programming approach does not directly provide an efficient method to find the policies.**

An iterative algorithm for policies with the simple form is proposed and investigated. The algorithm alternates between paging policy optimization and registration policy optimization. It finds a pair of individually optimal policies, but an example is given showing that the policies need not be jointly optimal.

I. Introduction

The growing demand for personal communication services is increasing the need for efficient utilization of the limited resources available for wireless communication. In order to deliver service to a mobile station (MS), the cellular network must be able to track the MS as it roams. In this paper, the problem of minimizing the cost of tracking is discussed. Two basic operations involved in tracking the MS are *paging* and *registration*.

There is a tradeoff between the paging and registration costs. If the MS registers its location within the cellular network more often, the paging costs are reduced, but the registration costs are higher. The traditional approach to paging and registration in cellular systems uses *registration areas* which are groups of cells [8]. An MS registers if and only if it changes registration area. Thus, when there is an incoming call directed to the MS, all the cells within its current registration area are paged. Another method uses *reporting centers* [2]. An MS registers only when it enters the cells of reporting centers, while every search for the MS is restricted to the vicinity of the reporting center to which it last reported.

Some dynamic registration schemes are examined in [3] : *time-based, movement-based*, and *distance-based*. Under such movements performed, or the distance traveled since the last location update. These policies are threshold policies and the thresholds depend on the MS motion activities. In [12], dynamic programming is used to determine an optimal *state-based* registration policy.

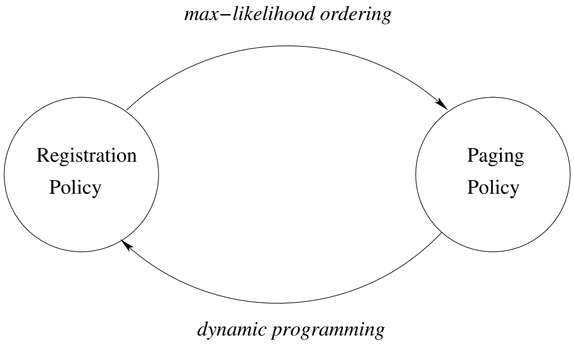

Fig. 1. Paging policy and registration policy generation

Basic paging policies can be classified as follows:

- *Serial Paging*. The cellular network pages the MS sequentially, one cell at a time.
- *Parallel Paging*. The cellular network pages the MS in a collection of cells simultaneously.

Serial paging policies have lower paging costs than parallel paging policies, but at the expense of larger delay. The method of parallel paging is to partition the cells in a service region into a series of indexed groups referred to as *paging areas*. When a call arrives for the MS, the cells in the first paging area are paged simultaneously in the first round and then,

if the MS is not found in the first round of paging, all the cells in the second paging area are paged, and so on. Given disjoint paging areas, searching them in the order of decreasing probabilities minimizes the the expected number of searches [14]. This paging order is denoted as the *maximum-likelihood serial paging order*. An interesting topic of paging is to design the optimal paging areas within delay constraints [11], [14], [17]. However, in this paper, we consider only serial paging polices.

Each paper mentioned above assumes a certain class of paging or registration policy. Given one policy (paging policy or registration policy) and the parameters of an assumed motion model, the counterpart policy (registration policy or paging policy, respectively) is found. For instance, the optimal

0-7803-7753-2/03/$17.00 (C) 2003 IEEE

paging policy is identified in [14] for a given registration policy. This is shown as the top branch of Figure 1. Conversely, an expanding "ping-pong" order paging policy suited to the given motion model is assumed in [12]. With this knowledge, dynamic programming is applied to solve for the optimal registration policy. This corresponds to the bottom branch of Figure 1.

Several studies have addressed minimizing the costs, considering the paging and registration policies together [1], [15], [16]. In [15], a timer-based registration policy combined with maximum-likelihood serial paging is introduced. The minimum paging cost can be represented by the distribution of locations where the MS last reported. Then an optimal timer threshold is selected to minimize the total cost of registration and paging. By contrast, a movement-based registration policy is used in [1]. An improvement of [15] is given in [16] by assuming that the MS knows not only the current time, but also its own state and the conditional distribution of its state given the last report. This is a state-based registration policy and is aimed to minimize the total costs by running a greedy algorithm on the potential costs. Although the papers discuss the registration and paging policy together, they don't consider the joint optimization of the registration policy and paging policy.

The structure of jointly optimal paging and registration policies is investigated in this paper. The conditional probability distribution of the states of an MS is viewed as a controlled Markov process, controlled by both the paging and registration polices at each time. The method of dynamic programming is applied, which in particular shows that the jointly optimal policies can be represented compactly by certain *reduced complexity laws* (RCLs). An iterative algorithm producing a pair of RCLs is proposed based on closing the loop in Figure 1. The algorithm is a heuristic which merges the approaches in [12] and [14].

The remainder of the paper is organized as follows. Notation and cost functions are introduced in Section II. Jointly optimal policies are investigated in Section III. The iterative optimization formula for computing individually optimal policy pairs is developed in Section IV. Examples and conclusions are given in Sections V and VI.

II. NETWORK MODEL

A. State description and cost

The motion of an MS is modeled by a discrete-time Markov process $(X(t) : t \geq 0)$ with finite state space S, one-step transition probability matrix $P = (p_{ij} : i, j \in S)$, and given initial state x_0. The state $i \in S$ that the MS is in at a given time determines the cell c that the MS is physically located in, and the state may indicate additional information such as the current velocity of the MS. Thus a cell c can be considered to be a set of one or more states, and the set $\mathcal{C}$ of all cells is a partition of S. It is assumed that the network knows the initial state x_0.

The state of the MS from one time t to the next evolves according to the given transition probabilities. The order of the possible events at a particular integer time instant $t \geq 1$ are as follows. First, it is announced whether the MS is to be paged, and the answer is "yes" with probability λ_p, independently of the state of the MS and all past events. The cost of the paging at time t is $\mathcal{P}N_t$, where $\mathcal{P}$ is the cost of searching one cell and N_t is the number of cells that are searched until the MS is found. As a result of being paged, the state of the MS is reported to the cellular network. Let $N_t = 0$ if the MS is not paged at time t. No paging or registration is considered for $t = 0$.

If the MS is not paged, then the MS decides whether to register. The cost of registration is $\mathcal{R}$ and the benefit of registration is that the cellular network learns the state of the MS. We say that a *report* occurs whenever either a paging or registration occurs, because in either case, the cellular network learns the state of the MS.

Let P_t denote the event that the MS is paged at time t, and let R_t denote the event that the MS registers at time t. For any set A, let I_A denote the indicator function of A, which is one on A and zero on the complement A^c. Probability vectors are considered to be row vectors. Given a state $l \in S$, let $\delta(l)$ denote the probability vector for S which assigns probability one to state l. Thus, $\delta_i(l) = I_{\{i=l\}}$. See the appendix for a review of the notions of σ algebras used in this paper.

B. Paging policy notation

For simplicity we consider only serial paging policies, so that cells are searched one at a time until the MS is located. It is also assumed that if the MS is present in the cell in which it is paged, it responds to the page successfully. In other words, no paging failure is allowed. It is further assumed that the time it takes to issue a single-cell page is negligible compared to one time step of the MS's motion model, so that paging is always successfully completed within one time step.

Let $\mathcal{N}_t$ denote the σ algebra representing the information available to the network by time t after the paging and registration decisions have been made and carried out. Thus, for $t \geq 0$,

$$\begin{aligned}\mathcal{N}_t &= \sigma((I_{P_s}, N_s, I_{R_s} : 1 \leq s \leq t),\\ &\qquad (X(s) : 1 \leq s \leq t \text{ and } I_{P_s \cup R_s} = 1))\end{aligned}$$

The initial state x_0 is treated as a constant, so even though it is known to the network it is not included in the definition of $\mathcal{N}_t$. Note that the initial σ algebra $\mathcal{N}_0$ is the trivial σ algebra: $\mathcal{N}_0 = \{\emptyset, \Omega\}$.

A paging policy u is a collection $u = (u(t) : t \geq 1)$ such that for each $t \geq 1$, $u(t)$ is an $\mathcal{N}_{t-1}$ measurable random variable with values in Perm($\mathcal{C}$), the set of permutations of the set of cells $\mathcal{C}$. Given a permutation a of $\mathcal{C}$, the *paging rank vector*, $r^a = (r^a_l : l \in \mathcal{S})$ is defined as follows. If cells are searched sequentially in the order specified by a, then r^a_l is the number of single-cell searches required to find the MS if it is in state l. For example, suppose $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$ and $\mathcal{C} = \{c_1, c_2, c_3\}$ with $c_1 = \{1, 2\}$, $c_2 = \{3, 4\}$, and $c_3 = \{5, 6\}$. Then if $a = (c_2, c_1, c_3)$, meaning to search cell c_2 first, c_1 second, and c_3 third, then $(r^a_1, \ldots, r^a_6) = (2, 2, 1, 1, 3, 3)$.

C. Registration policy notation

Let $\mathcal{M}_t$ denote the σ algebra representing the information available to the MS by time t, after the paging and registration decisions for time t have been made and carried out. Thus,

$$\mathcal{M}_t = \sigma(X(s), I_{P_s}, N_s, I_{R_s} : 1 \le s \le t).$$

The MS also knows the initial position x_0, which is treated as a constant. In practice an MS wouldn't learn N_s, the number of pages used to find the MS at time s. While we assume such information is available to the MS, we will see that optimal policies need not make use of the information. With this definition, we have $\mathcal{N}_t \subset \mathcal{M}_t$, meaning that the MS knows everything the network knows (and typically more).

When the MS has to decide whether to register at time t, it already has the information $\mathcal{M}_{t-1}$. In addition it knows $X(t)$ and I_{P_t}. If the MS is paged at time t, then the network learns the state of the MS as a result, so there is no advantage for the MS to register at time t. Thus, we assume without loss of generality that the MS does not register at time t if it is paged at time t. This leads to the following definition.

A registration policy v is a collection $v = (v(t) : t \ge 1)$ such that for each $t \ge 1$, $v(t)$ is an $\mathcal{M}_{t-1}$ measurable random vector with values in $[0,1]^S$ with the following interpretation. Given the information $\mathcal{M}_{t-1}$, if $X(t) = l$ and if the MS is not paged at time t, then the MS registers with probability $v_l(t)$.

D. Cost function

Let β be a number with $0 < \beta < 1$, called the discount factor. One interpretation of β is that $c\beta^t$ is the effective cost at time zero of cost c to be paid t time units in the future. Another interpretation is that $1/(1-\beta)$ is the rough time horizon of interest. Given a paging policy u and registration policy v, the expected infinite horizon discounted cost $C(u,v)$ is defined as

$$C(u,v) = E\left[\sum_{t=1}^{\infty} \beta^t \{\mathcal{P} I_{P_t} N_t + \mathcal{R} I_{R_t}\}\right]. \qquad (1)$$

The pair (u,v) is *jointly optimal* if $C(u,v) \le C(u',v')$ for all other paging policies u' and registration policies v'.

III. JOINTLY OPTIMAL POLICIES

This section investigates the structure of jointly optimal policies by using the theory of dynamic programming for Markov control problems with partially observed states. While the structure results do not directly yield a computationally feasible solution, they shed light on the nature of the problem. In particular it is found that there are jointly optimal policies (u,v) such that, for each t, $u(t)$ and $v(t)$ are functions of the amount of time elapsed since the last report and the last reported state.

Intuitively, the paging policies are selected based on the past of the registration policy, because the past of the registration policy influences the conditional distribution of the MS state. On the other hand, by the nature of dynamic programming, the optimal choice of registration policy at a given time depends on future costs, which are determined by the future of the registration policy. To break this cycle, we consider the problem more from the viewpoint of the network. In order that current decisions not depend on past actions, the state space is augmented by the conditional distribution of the state of the MS given the information available to the network.

A. Evolution of conditional distributions

For $t \ge 0$, let $w(t)$ be a row vector equal to the conditional probability distribution of $X(t)$, given the observations available to the network up to time t (including the outcomes of a report at time t, if there was any). That is, $w_j(t) = P[X(t) = j|\mathcal{N}_t]$ for $j \in S$. Note that with probability one, $w(t)$ is a probability vector with index set $\mathcal{S}$. Intuitively, the network can control the distribution valued process $(w(t))$ by dictating the registration policy of the MS. Since $\mathcal{N}_0$ is the trivial σ algebra and $X(0) = x_0$, the initial conditional distribution is given by $w(0) = \delta(x_0)$.

While the network may not know the recent past trajectory of the state process, it can still estimate the registration policies used by the MS. Such estimates play a role in how the network can recursively update the $w(t)$'s. Thus, let $\hat{v}_j(t) = E[v_j(t)|\mathcal{N}_{t-1}]$. Note that $\hat{v} = (\hat{v}(t) : t \ge 1)$ is itself a registration policy. Indeed, since $\mathcal{N}_{t-1} \subset \mathcal{M}_{t-1}$, $\hat{v}(t)$ is $\mathcal{M}_{t-1}$ measurable, and it takes values in $[0,1]^S$. If $\hat{v}$ is used as a registration policy by the MS, the MS's actual decision of whether to register at time t depends on both $\hat{v}(t)$ and $X(t)$. Since the network may not know $X(t)$, the network can't itself know exactly when the MS will register, even though the network does know $\hat{v}(t)$ at time $t-1$.

Define a function Φ as follows. Let w be a probability distribution on S and let $b \in [0,1]^S$. Let $\Phi(w,b)$ denote the probability vector on S defined by

$$\Phi_l(w,b) = \frac{\sum_{j\in S} w_j p_{jl}(1-b_l)}{\sum_{l'\in S}\sum_{j\in S} w_j p_{jl'}(1-b_{l'})}.$$

$\Phi(w,b)$ is undefined if the denominator in this definition is zero. The meaning of Φ is that if at time t the network knows that $X(t)$ has distribution w, if no paging occurs at time $t+1$, and if the MS registers at time $t+1$ with probability $b_{X(t+1)}$, then $\Phi(w,b)$ is the conditional distribution of $X(t+1)$ given no registration occurs at time $t+1$. This interpretation is made precise in the next lemma. The proof is omitted.

Lemma 3.1: The following holds:

$$\begin{aligned} w(t+1) &= \delta(X(t+1)) I_{P_{t+1}\cup R_{t+1}} \\ &\quad + \Phi(w(t), \hat{v}(t+1)) I_{P^c_{t+1}\cap R^c_{t+1}}. \end{aligned}$$

B. New state process

For $t \ge 1$, let $\Theta(t) = (w(t), I_{P_t}, N_t, I_{R_t})$. Note that the tth term in the cost function is a function of $\Theta(t)$. Note also that $\Theta(t)$ is measurable with respect to $\mathcal{N}_t$, so that the network can calculate $\Theta(t)$ at time t (after possible paging and registration). Moreover, the first coordinate of $\Theta(t)$, namely $w(t)$, can be updated with increasing t with the help of Lemma 3.1. The random process $(\Theta(t) : t \ge 0)$ can be viewed as a controlled Markov process, adapted to the family of σ algebras $(\mathcal{N}_t : t \ge 0)$ with controls $(u(t), \hat{v}(t) : t \ge 1)$. Note that $u(t+1)$ and $\hat{v}(t+1)$ are each $\mathcal{N}_t$ measurable for each $t \ge 0$. The one step transition probabilities for $(\Theta(t))$ are given as follows. (The variables j and l range over the set of states S.)

$\Theta(t+1)$	$Probability$
$(\delta(l), 1, r_l^{u(t+1)}, 0)$	$\lambda_p \sum_j w_j(t) p_{jl}$
$(\delta(l), 0, 0, 1)$	$(1-\lambda_p) \sum_j w_j(t) p_{jl} \hat{v}_l(t+1)$
$(\Phi(w(t), \hat{v}(t+1)), 0, 0, 0)$	$(1-\lambda_p) \sum_j w_j(t) p_{jl} (1-\hat{v}_l(t+1))$

Since $\mathcal{N}_t \subset \mathcal{M}_t$ (i.e. the MS knows at least as much as the network) the set of registration policies that are adapted to $(\mathcal{N}_t : t \geq 0)$ is a subset of all the registration policies. However, we have shown that for an arbitrary registration policy v there exists a registration policy $\hat{v}$ that is adapted to the family of σ algebras $(\mathcal{N}_t : t \geq 0)$ which yields the same cost as v. Therefore, without loss of generality, we can restrict attention to registration policies $\hat{v}$ that are adapted to $(\mathcal{N}_t : t \geq 0)$.

Combining the observations summarized in this section, we arrive at the following proposition.

Proposition 3.1: The original joint optimization problem is equivalent to a Markov optimal control problem with state process $(\Theta(t) : t \geq 0)$ adapted to the family of σ algebras $(\mathcal{N}_t : t \geq 0)$, with controls $(u(t), \hat{v}(t) : t \geq 1)$.

C. Dynamic programming equations

Above it was assumed the $w(0) = \delta(x_0)$ where x_0 is the initial state of the MS, assumed known by the network. In order to apply the dynamic programming technique, in this section the initial distribution $w(0)$ is allowed to be any probability distribution on S. It is assumed that the network knows $w(0)$ at time zero, and that the initial state of the MS is random, with distribution $w(0)$. The evolution of the system as described in the previous section is well defined for an arbitrary initial distribution $w(0)$. Let E_w denote conditional expectation in case the initial distribution $w(0)$ is taken to be w. The initial σ algebra $\mathcal{N}_0$ is still the trivial σ algebra, since $w(0)$ is treated as a given constant.

Define the cost with n steps to go as

$$U_n(w) = \min_{u,\hat{v}} E_w[\sum_{t=1}^{n} \beta^t \{\mathcal{P} I_{P_t} N_t + \mathcal{R} I_{R_t}\}]$$

Next apply the backwards solution method of dynamic programming, by separating out the $t = 1$ term in the cost for $n+1$ steps to go. This yields

$$\begin{aligned} U_{n+1}(w) \quad = \quad & \min_{u,\hat{v}} \beta \Bigg[\lambda_p \mathcal{P} \sum_j \sum_l w_j p_{jl} r_l^{u(1)} \\ & + (1-\lambda_p) \mathcal{R} \sum_j \sum_l w_j p_{jl} \hat{v}_l(1) \\ & + E_w[E_w[\sum_{t=2}^{n+1} \beta^{t-1} \{\mathcal{P} I_{P_t} N_t + \mathcal{R} I_{R_t}\} | \mathcal{N}_1]] \Bigg] \end{aligned}$$

Note that $u(1)$ and $\hat{v}(1)$ are both measurable with respect to the trivial σ algebra $\mathcal{N}_0$. Therefore these controls are constants. Henceforth we write b for the registration decision vector $\hat{v}(1)$. The vector b ranges over the space $[0,1]^S$.

The first sum in the expression for $U_{n+1}(w)$ involves the control policies only through the choice of the permutation $u(1)$. This sum is simply the mean number of single-cell pages required to find the MS given that the state of the MS has distribution given by the product wP, where P is the matrix of state transition probabilities. It is well known that the optimal search order is to first search the cell with the largest probability, then search the cell with the second largest probability, and so on [14]. Ties can be broken arbitrarily. The first sum in the expression for $U_{n+1}(w)$ can thus be replaced by $s(wP)$, where $s(q)$ denotes the mean number of single cell pages required to find the MS given that the state of the MS has distribution q and the optimal paging policy is used.

The dynamic programming equation thus becomes

$$\begin{aligned} U_{n+1}(w) &= \beta \lambda_p \mathcal{P} s(wP) \\ & \min_b \beta \Bigg[\sum_j \sum_l w_j p_{jl} \{\lambda_p U_n(\delta(l)) \\ & \quad + (1-\lambda_p) b_l (\mathcal{R} + U_n(\delta(l)))\} \\ & + \left(\sum_j \sum_l w_j p_{jl} (1-b_l) \right) U_n(\Phi(w,b)) \Bigg] . \end{aligned} \tag{2}$$

Formally we denote this equation as $U_{n+1} = T(U_n)$. By a standard argument for dynamic programming with discounted cost, T has the following contraction property:

$$\sup_w |T(U) - T(U')| \leq \beta \sup_w |U - U'| \tag{3}$$

for any bounded, measurable functions U and U', defined on the space of all probability distributions w on S. Consequently [4], [5], there exists a unique U_* such that $T(U_*) = U_*$, and $U_n \to U$ uniformly as $n \to \infty$. Moreover, U_* is the minimum possible cost, and a jointly optimal pair of paging and registration policies is given by a pair $(\bar{f}, \bar{g})$ of state feedback controls, for the state process $(w(t))$. A jointly optimal control is given by $u(t) = \bar{f}(w(t-1))$ and $v(t) = \bar{g}(w(t-1))$, where $\bar{f}$ and $\bar{g}$ are determined as follows. For any probability distribution w on S, $\bar{f}(w)$ is the permutation listing the states of S in order of decreasing values of wP, and $\bar{g}(w)$ is a value of b that achieves the minimum in the right hand side of (2) with U_n replaced by U_*. Then if there is no report at time $t+1$, the conditional distribution $w(t)$ is updated simply by:

$$w(t+1) = \Phi(w(t), \bar{g}(w(t))) \tag{4}$$

Clearly under such stationary state feedback control laws $(\bar{f}, \bar{g})$, the process $(w(t) : t \geq 0)$ is a time-homogeneous Markov process. Note that the optimal mapping $\bar{f}$ does not depend on $\bar{g}$.

Lemma 3.2: The registration policy $\bar{g}$ can be taken to be $\{0,1\}^S$ valued (rather than $[0,1]^S$ valued) without loss of optimality.

Proof: It is first proved that U_n is concave for any given $n \geq 0$. Suppose w_1 and w_2 are two probability distributions on S, suppose $0 < \eta < 1$ and suppose $w = \eta w_1 + (1-\eta)w_2$. Then $U_n(w)$ can be viewed as the cost to go given the MS has distribution w_1 with probability η and distribution w_2 with probability $1-\eta$, and the network does not know which distribution is used. The sum $\eta U_n(w_1) + (1-\eta)U_n(w_2)$ has a similar interpretation, except the network does know which distribution is used. Thus, the sum is less than or equal to $U_n(w)$, so that U_n is concave. Therefore U_* is also concave.

Given a function H defined on the space of all probability vectors for S, let $\tilde{H}$ be an extension of H defined on the positive quadrant R_+^S as follows. For any probability vector w and any constant $c \geq 0$, $\tilde{H}(cw) = cH(w)$. It is easy to show that if H is concave then the extension $\tilde{H}$ is also concave. With this notation, the dynamic programming equation for U_* can be written as:

$$\begin{aligned} U_*(w) &= \beta\lambda_p \mathcal{P}s(wP) \\ &\min_b \beta \Bigg[\sum_j \sum_l w_j p_{jl} \left\{ \lambda_p U_*(\delta(l)) \right. \\ &\quad \left. +(1-\lambda_p) b_l (\mathcal{R} + U_*(\delta(l))) \right\} \\ &\quad + \tilde{U_*}(wPdiag(1-b)) \Bigg]. \end{aligned}$$

where $diag(1-b)$ is the diagonal matrix with lth entry $1-b_l$. The expression to be minimized over b in this equation is a concave function of b, and hence the minimum of the function occurs at one of the extreme points of $[0,1]^S$, which are just the binary vectors $\{0,1\}^S$. The minimizing b is $\bar{g}(w)$. This completes the proof of the lemma. ■

D. Reduced complexity laws

Given a pair of feedback controls $(\bar{f}, \bar{g})$, a more compact representation of the controls is possible. Indeed, suppose the controls are used, and suppose in addition that $X(0) = x_0$, where x_0 is an initial state known to the network. Given $t \geq 1$, define $k \geq 1$ and $i_0 \in S$ as follows. If there was a report before time t, let $t-k$ be the time of the last report before t. If there was no report before time t let $k = t$. In either case, let $i_0 = X(t-k)$. Since the network knows $X(t-k)$ at time $t-k$ (after possible paging or registration), we have that $w(t-k) = \delta(i_0)$. Since there were no state updates during the times $t-k+1, \ldots, t-1$, it follows that $w(t-1)$ is the result of applying the update (4) $k-1$ times, beginning with $\delta(i_0)$. Hence, $w(t-1)$ is a function of i_0, k. Moreover, since $u(t) = \bar{f}(w(t-1))$ and $v(t) = \bar{g}(w(t-1))$, it follows that both the permutation $u(t)$ and the registration decision vector $v(t)$ are determined by i_0 and k. Let f and g denote the mappings such that $u(t) = f(i_0, k)$ and $v(t) = g(i_0, k)$. Note that $f(i_0, k)$ is a permutation of S and $g(i_0, k) \in \{0,1\}^S$ for each i_0, k. We call the mappings f, g *reduced complexity laws* (RCLs). We have shown the following proposition.

Proposition 3.2: There is no loss in optimality for the original joint paging and registration problem to use policies based on RCLs.

Figure 2 shows an example of a registration RCL g for a three-state Markov chain. The augmented state of a MS is a triple (i_0, k, j), such that i_0 is the state at the time of the last report, k is the elapsed time since the last report, and j is the current state. Augmented states marked with an "x" are those for which $g_j(i_0, k) = 1$, meaning that registration occurs (if paging doesn't occur first). An MS traverses a path from left to right until either it is paged, or until it hits a state marked with an "x", at which time its augmented state instantaneously jumps. The figure shows the path of a MS that began in augmented state $(i_0, k, j) = (2, 0, 2)$. At relative time $k = 5$ the MS entered state 3, hitting an "x", causing the extended state to instantly change to $(3, 0, 3)$. Three time units after that, upon entering state 1, the MS is paged. This causes the augmented state to instantly jump to $(1, 0, 1)$.

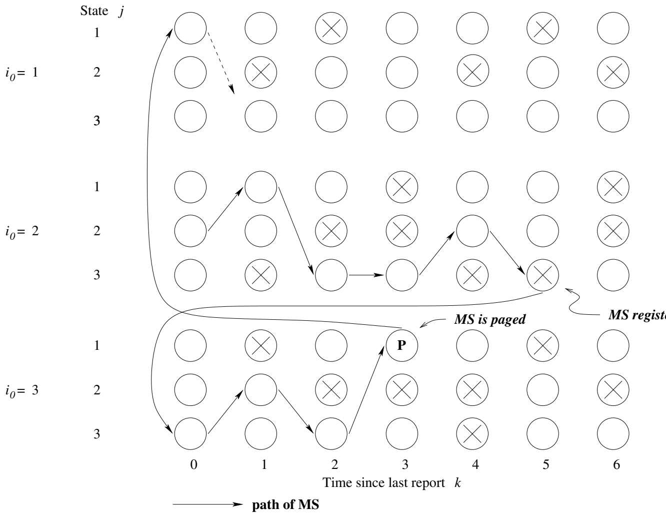

Fig. 2. Example registration policy for three-state Markov chain

IV. ITERATIVE ALGORITHM FOR FINDING INDIVIDUALLY OPTIMAL POLICIES

A. Overview of iterative optimization formulation

While jointly optimal policies can be efficiently represented by RCLs f and g, the dynamic program method described for finding the optimal policies is far from computationally feasible, even for small state spaces, because functions of distributions on the state space must be considered. In this section we explore the following method for finding a pair of policies with a certain local optimality property. First it is show how to find, for a given paging RCL f, an optimal registration RCL g. Then it is shown how to find, for a given registration RCL g, an optimal paging RCL f. Iterating between these two optimization problems produces a pair of RCL policies (f, g) such that for each policy fixed, the other is optimal. Such pairs of RCL policies are said to be *individually optimal.*

In this section we impose the constraint that a MS must register if $k \geq k_{max}$, for some large integer constant k_{max}. With this constraint, the sets of possible registration and paging RCLs are finite, and numerical computation is feasible for fairly large state spaces. The initial state x_0 is assumed to be known and we write $C(f, g)$ for the averaged infinite horizon, discounted cost, for paging RCL f and registration RCL g.

B. Optimal registration RCL for given paging RCL

Suppose a paging RCL f is fixed. In this subsection we address the problem of finding a registration RCL g that minimizes $C(f,g)$ with respect to g. Dynamic programming is again used, but here the viewpoint of the MS is taken. The states used for dynamic programming in this section are the augmented states of the form (i_0,k,j), rather than the set of all probability distributions on S.

Since time is implicitly included in the variable k in the augmented state, it is computationally more efficient to consider dynamic programming iterations based on cycles rather than on single time steps, where each cycle ends when there is a report. Let τ_m be the time of the m^{th} report. Replacing the infinite horizon by time horizon τ_m reduces $C(f,g)$ to

$$E\left[\sum_{t=1}^{\tau_m}\beta^t\left\{\mathcal{P}I_{P_t}N_t+\mathcal{R}I_{R_t}\right\}\right] \tag{5}$$

Letting $m\to\infty$ in (5) yields $C(f,g)$.

Then for each (i_0,j,k), write $V_m(i_0,k,j)$ for the *cost-to-go* for $m\geq 1$ update cycles:

$$V_m(i_0,k,j)=\min_u E\left[\sum_{t=1}^{\tau_m}\beta^t\left\{\mathcal{P}I_{P_t}N_t+\mathcal{R}I_{R_t}\right\}\right], \tag{6}$$

where the expectation E is taken assuming that (a) the paging RCL f is used for the paging policy, (b) at $t=0$ the MS is in state j, and (c) the last report occurred k time units earlier in state i_0. Also, define $V_0(i_0,k,j)\equiv 0$ since the cost is zero when there are no report cycles to go.

The dynamic programming optimality equations are given by

$$\begin{aligned}V_m(i_0,k,j)=\beta\sum_{l\in\mathcal{S}}p_{jl}&\left[\lambda_p(\mathcal{P}r_l^{f(i_0,k+1)}+V_{m-1}(l,0,l))\right.\\&\left.+\,(1-\lambda_p)\min\left\{V_m(i_0,k+1,l),\mathcal{R}+V_{m-1}(l,0,l)\right\}\right]\end{aligned} \tag{7}$$

As mentioned earlier, registration is forced at relative time $k=k_{max}+1$ for some large but fixed value k_{max}. Therefore we set $V_m(i_0,k_{max}+1,l)=\infty$ and use (7) only for $0\leq k\leq k_{\max}$. These equations represent the basic dynamic programming optimality relations. For each possible next state, the MS chooses whichever action has lesser cost: either continuing the current registration cycle or registering for cost $\mathcal{R}$.

Equation (7) can be used to compute the functions V_m sequentially in m as follows. The initial conditions are $V_0\equiv 0$. Once V_{m-1} is computed, the values $V_m(i_0,k,j)$ can be computed using (7), sequentially for k decreasing from $k_{\max}$ to 0. Formally we denote this computation as $V_m=T(V_{m-1})$. The mapping T is a contraction with constant β in the sup norm, so that V_m converges uniformly to a function V_* satisfying the limiting form of (7):

$$\begin{aligned}V_*(i_0,k,j)=\beta\sum_{l\in\mathcal{S}}p_{jl}&\left[\lambda_p(\mathcal{P}r_l^{f(i_0,k+1)}+V_*(l,0,l))\right.\\&\left.+\,(1-\lambda_p)\min\left\{V_*(i_0,k+1,l),\mathcal{R}+V_*(l,0,l)\right\}\right]\end{aligned} \tag{8}$$

for $0\leq k\leq k_{max}$, and $V_*(i_0,k_{max}+1,l)\equiv\infty$. The corresponding optimal registration $RCL\ g^*$ is given by

$$g_l^*(i_0,k)=\begin{cases}0, & \text{if } V_*(i_0,k,l)\leq V_*(l,0,l)+\mathcal{R}\\ 1, & \text{else.}\end{cases} \tag{9}$$

for $i_0\in S$ and $1\leq k\leq k_{max}$.

Thus, for a given paging RCL f, we have identified how to compute a registration RCL g to minimize $C(f,g)$.

C. Optimal paging RCL for given registration RCL

Suppose a registration RCL g is fixed. In this subsection we address the problem of finding a paging RCL f to minimize $C(f,g)$. For $i_0\in S$ and $0\leq k\leq k_{max}$, let $w(i_0,k)$ denote the conditional probability distribution of the state of the MS, given that the most recent report occurred k time units earlier and the state at the time of the most recent report was i_0. Thus, $w(i_0,0)=\delta(i_0)$, and for larger k the w's can be computed by the recursion:

$$w(i_0,k+1)=\Phi(w(i_0,k),g(i_0,k))$$

The permutation $f(i_0,k)$ is simply the permutation of states to be used when the MS must be paged k time units after the previous report. At such time the conditional distribution of the state of the MS given the observations of the base station is $Pw(i_0,k-1)$. Thus, the probability the MS is located in cell c, just before the paging begins is given by

$$p(c|i_0,k)=\sum_{j\in S}\sum_{l\in c}w_j(i_0,k-1)p_{jl}$$

Finally, $f(i_0,k)$ is simply the permutation ordering the cells c according to decreasing values of the probabilities $p(c|i_0,k)$.

D. Iterative optimization algorithm

In the previous subsections we described how to find an optimal g for given f and vice versa. This suggests an iterative method for finding an individually optimal pair (f,g). The method works as follows. Fix an arbitrary registration RCL g^0. Then execute the following steps.

- Find a paging RCL f^0 to minimize $C(f^0,g^0)$
- Find a registration RCL g_1 to minimize $C(f^0,g^1)$,
- Find a paging RCL f^1 to minimize $C(f^1,g^1)$, and so on.

Then $C(f^0,g^0)\geq C(f^0,g^1)\geq C(f^1,g^1)\geq C(f^1,g^2)\geq\cdots$ Since there are only finitely many RCL policies, it must be that for some integer d, $C(f^d,g^d)=C(f^d,g^{d+1})$. By construction, the paging RCL f^d is optimal given the registration RCL g^d. Similarly, g^{d+1} is optimal given f^d. However, since $C(f^d,g^d)=C(f^d,g^{d+1})$, it follows that g^d is also optimal given the registration policy f^d. Therefore, (f^d,g^d) is an individually optimal pair of RCL policies.

V. EXAMPLES

A. Rectangular grid example

In the rectangular grid topology, each cell has four neighbors. The diagram to the left in Figure 3 shows the finite $i_{max}\times j_{max}$ rectangular grid topology. To provide the full complement of four neighbors to cells on the edges of the

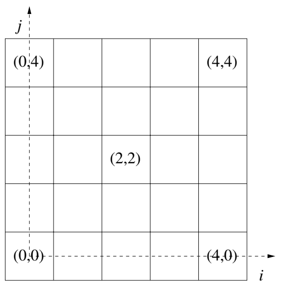

Fig. 3. Rectangular grid motion model

grid, the region is wrapped into a torus. The "wrapped" finite grid can serve to approximate larger sets of cells. Also, by the symmetry of the torus, the functions $f(i_0, k), g(i_0, k)$ and distributions $w(i_0, k)$ need be computed for only one value of last reported cell i_0. Each cell in Figure 3 is represented by the index pair (i, j), where $i = 0, 1, \ldots, i_{max} - 1$ is the index for the horizontal axis, and $j = 0, 1, \ldots, j_{max} - 1$ is the index for the vertical axis.

As an example, consider the 15×15 rectangular grid motion model with motion parameters $p_{sty} = 0.4$, $p_u = p_d = p_l = 0.1$, $p_r = 0.3$, $x_0 = (5, 5)$ and other parameters $\lambda_p = 0.03$, $\mathcal{P} = 1$, $\mathcal{R} = 0.6$, $\beta = 0.9$, and $k_{max} = 200$. Figure 4 shows for selected times t the state $X(t)$, indicated by a small black square, and the conditional state distribution $w(t)$, indicated as a moving bubble. The distribution $w(t)$ collapses to a single unit mass point at $t = 9$ due to a page and at $t = 27$ due to a registration. Roughly speaking, the MS registers when it is not where the network expects it to be, given the last report received by the network. For instance, at time $t = 26$ the MS is located at the tail edge of the bubble, so the network has low accuracy in guessing the MS location. One time unit later, at t=27, the MS finds itself so far from where the network thinks it should be that the MS registers.

B. Simple Example

The following is an example of a small network for which jointly optimal paging and registration policies can be computed. The example also affords a pair of individually optimal RCLs which are not jointly optimal. The space structure of the example is shown in Figure 5. $S = \{0, 1, 2, 3, 4\}$ and $\mathcal{C} = \{c_0, c_1, c_2\}$ with $c_0 = \{0\}$, $c_1 = \{1, 2\}$, $c_2 = \{3, 4\}$. From state 0, the MS transits to state 1 with probability 0.4 and to state 3 with probability 0.6. The other possible transitions shown in the figure have probability 1. The initial state is taken to be 0.

We first describe the jointly optimal pair of paging and registration policies. We consider, without loss of optimality, policies given by feedback control laws ($\bar{f}$,$\bar{g}$) as described in Section III. Thus we take $u(t) = \bar{f}(w(t-1))$ and $v(t) = \bar{g}(w(t-1))$. Due to the special structure of this example, the process $w(t)$ takes values in a set of at most seven states, and the possible transitions are shown in Figure 6. The dynamic programming problem for jointly optimal policies thus reduces to a finite state problem. The optimal choice of the mapping $\bar{f}$

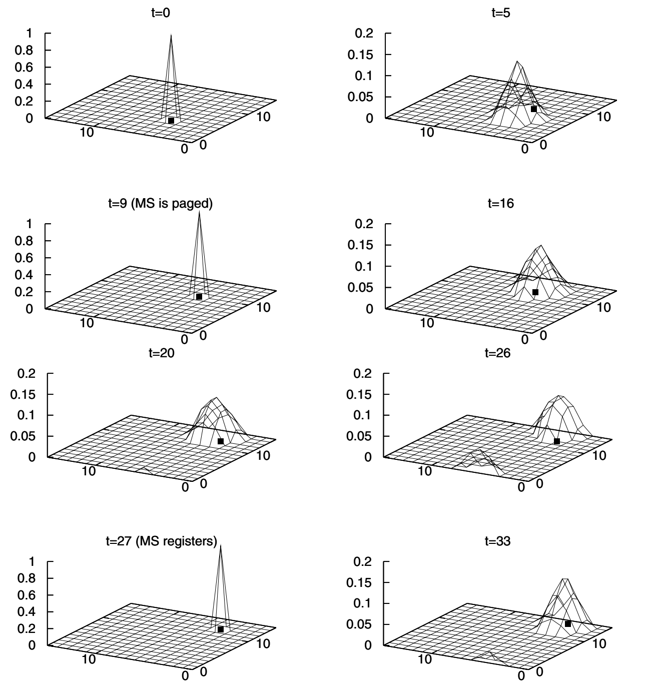

Fig. 4. Evolution of the state $X(t)$ and the conditional distribution of the state $w(t)$ for the rectangular grid example.

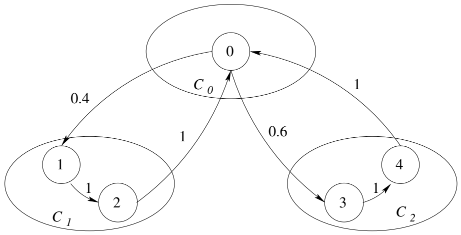

Fig. 5. Simple example

is given by $\bar{f}^*(w)$, which pages states in decreasing order of wP. It remains to find the optimal registration policy mapping $\bar{g}$.

We claim that if $t \; mod \; 3 = 0$ or $t \; mod \; 3 = 2$, then it is optimal to not register at time t. Indeed, if $t \; mod \; 3 = 0$ then the network already knows the MS is in state 0, so registration would cost $\mathcal{R}$ and provide no benefit. If $t \; mod \; 3 = 2$, then the network knows that the MS will be in state 0 at time $t + 1$, which is the next time of a potential page. Thus, again the registration at time t would cost $\mathcal{R}$ and provide no benefit. This proves the claim.

Therefore, it remains to find the optimal registration vector

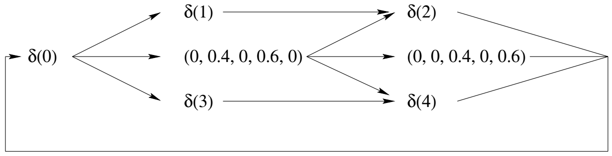

Fig. 6. Evolution of $w(t)$ for the simple example.

TABLE I

REGISTRATION POLICIES $\bar{g}_A$, $\bar{g}_B$,$\bar{g}_C$, $\bar{g}_D$ FOR THE SIMPLE EXAMPLE.

Policy	$\bar{g}(\delta(0))$	$P\left[R_1 \vert P_1^c\right]$	$P\left[N_2 = 2 \vert P_1^c \cap P_2\right]$
A	$(0,0,0,0,0)$	0	0.4
B	$(0,1,0,0,0)$	0.4	0
C	$(0,0,0,1,0)$	0.6	0
D	$(0,1,0,1,0)$	1	0

$v(t)$ to use when $t \ mod \ 3 = 1$. Such vector is deterministic, given by $\bar{g}(\delta(0))$. There are essentially only four possible choices for $\bar{g}(\delta(0))$, as indicated in Table I.

The cost for any pair ($\bar{f}$,$\bar{g}$) is given by

$$C(\bar{f},\bar{g}) = \frac{\mathcal{R}\beta(1-\lambda_p)P\left[R_1|P_1^c\right]}{1-\beta^3} + \frac{\lambda_p\mathcal{P}(1.4\beta+\beta^2+\beta^2(1-\lambda_p)P\left[N_2 = 2|P_1^c \cap P_2\right]+\beta^3)}{1-\beta^3}$$

Consulting Table I we thus find that ($\bar{f}^*$, $\bar{g}_A$) is jointly optimal if $\mathcal{R} \geq \lambda_p\mathcal{P}\beta$, and ($\bar{f}^*$, $\bar{g}_B$) is jointly optimal if $\mathcal{R} \leq \lambda_p\mathcal{P}\beta$.

For the remainder of this example we consider policies given by RCLs. Under the assumption that $0 < \mathcal{R} \leq \lambda_p\mathcal{P}\beta$, the pair of mappings ($\bar{f}^*$, $\bar{g}_B$) is equivalent to a pair of RCLs, which we denote by (f_B, g_B). Under g_B, the MS registers only after entering state 1 and not being paged. The pair (f_B,g_B) is jointly optimal, and hence it is also individually optimal. Similarly, let (f_C,g_C) be RCLs corresponding to the feedback mappings ($\bar{f}^*$,$\bar{g}_C$). In particular, an MS using registration RCL g_C registers only after entering state 3 and not being paged.

Proposition 5.1: (f_C,g_C) is individually optimal, but not jointly optimal.

Proof: The paging RCL f_C is optimal for the registration RCL g_C because for g_C fixed, it is equivalent to the optimal feedback mapping $\bar{f}^*$. Suppose then that the MS uses the paging RCL f_C. Note that if the MS does not report at time $t = 1$, and if it is paged at time $t = 2$, the network will page cell c_1 first. Hence, if the MS enters state 3 at time $t = 1$ and if it is not paged at $t = 1$, then by registering for cost $\mathcal{R}$ it can avoid the two or more pages required at time $t = 2$ in case of a page at $t = 2$. Since $\mathcal{R} \leq \lambda_p\mathcal{P}\beta$, it is optimal to have the MS register at $t = 1$ in this situation. Thus g_C is optimal for f_C, so the pair is individually optimal. However, (f_C,g_C) has the same cost as ($\bar{f}^*$,$\bar{g}_C$). Thus, (f_C,g_C) is not jointly optimal. ■

VI. CONCLUSIONS

In this paper, we show how the joint paging and registration optimization problem can be formulated as a dynamic programming problem with partially observed states. In addition, an iterative method is proposed, involving dynamic programming with a finite state space, in order to find individually optimal pairs of RCLs. A possible direction of future research is to apply approximation methods such as neuro-dynamic programming [6], in order to cope with very large numbers of states for the iterative method, or to derive other approaches for solving the joint paging and registration problem.

VII. APPENDIX

Some basic definitions involving σ algebras are collected in this appendix. In this paper the network only observes random variables with finite numbers of possible outcomes, so that emphasis is given to conditioning with respect to finite σ algebras.

The collections of random variables considered in this paper are defined on some underlying probability space. A probability space is a triple $(\Omega, \mathcal{F}, P)$, such that Ω is the set of all possible outcomes, $\mathcal{F}$ is a σ algebra of subsets of Ω (so $\emptyset \in \mathcal{F}$ and $\mathcal{F}$ is closed under complements and countable intersections) and P is a probability measure, mapping each element of $\mathcal{F}$ to the interval $[0, 1]$. The sets in $\mathcal{F}$ are called events. A random variable X is a function on Ω which is $\mathcal{F}$ measurable, meaning that $\mathcal{F}$ contains all sets of the form $\{\omega : X(\omega) \leq c\}$. In the remainder of this section, $\mathcal{N}$ denotes a σ algebra that is a subset of $\mathcal{F}$. Intuitively, $\mathcal{N}$ models the information available from some measurement: one can think of $\mathcal{N}$ as the set of events that can be determined to be true or false by the measurement. A random variable Y is said to be $\mathcal{N}$ measurable if $\mathcal{N}$ contains all sets of the form $\{\omega : Y(\omega) \leq c\}$. Intuitively, Y is $\mathcal{N}$ measurable if the information represented by $\mathcal{N}$ determines Y.

An atom B of $\mathcal{N}$ is a set $B \in \mathcal{N}$ such that if $A \subset B$ and $A \in \mathcal{N}$ then either $A = \emptyset$ or $A = B$. Note that if $C \in \mathcal{N}$ and B is an atom of $\mathcal{N}$, then either $B \subset C$ or $B \subset C^c$. If $\mathcal{N}$ is finite (has finite cardinality) then there is a finite set of atoms $B_1, \ldots, B_m$ in $\mathcal{N}$ such that each element of $\mathcal{N}$ is either $\emptyset$ or the union of one or more of the atoms.

Given a random variable X with finite mean, one can define $E[X|\mathcal{N}]$ in a natural way. It is an $\mathcal{N}$ measurable random variable such that $E[XZ] = E[E[X|\mathcal{N}]Z]$ for any bounded $\mathcal{N}$ measurable random variable Z. In particular, if A is an atom in $\mathcal{N}$, then $E[X|\mathcal{N}]$ is equal to $E[XI_A]/P[A]$ on the set A. (Any two versions of $E[X|\mathcal{N}]$ are equal with probability one.)

Given a random variable Y, we write $\sigma(Y)$ as the smallest σ algebra containing all sets of the form $\{\omega \in \Omega : Y(\omega) \leq c\}$. The notation $E[X|Y]$ is equivalent to $E[X|\sigma(Y)]$. In case Y is a random variable with a finite number of possible outcomes $\{y_1, \ldots, y_m\}$, the σ algebra $\sigma(Y)$ is finite with atoms $B_i = \{\omega : Y(\omega) = y_i\}$, $1 \leq i \leq m$. Furthermore, given a random variable X with finite mean, $E[X|Y]$ is the function on Ω which is equal to $\frac{E[XI_{B_i}]}{P[B_i]}$ on B_i for $1 \leq i \leq m$.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under grant NSF ITR 00-85929, and the Army Research Office under grant DAAH04-95-1-0246.

REFERENCES

[1] I.F. Akyildiz, J.S.M. Ho, and Y.B Lin, “Movement-based location update and selective paging for PCS networks,” *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629–638, August 1996.

[2] A. Bar-Noy and I. Kessler, “Tracking mobile users in wireless communication networks,” *Proc. IEEE Infocom*, pp. 1232–1239, 1993.

[3] A. Bar-Noy, I. Kessler and M. Sidi, “Mobile users: To update or not to update?” *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629–638, August 1996.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, Belmont, MA. 1995.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II.* Athena Scientific, Belmont, MA. 1995.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA. 1996.

[7] Y. Birk and Y. Nachman, "Using direction and elapsed-time information to reduce the wireless cost of locating mobile units in cellular networks." *Wireless Networks*, 1995.

[8] V. K. Garg and J. E. Wilkes, *Wireless and Personal Communications Systems*, Prentice Hall, New Jersey, 1996.

[9] C. -L. I, G. P. Pollini, and R. D. Gitlin, "Optimum location area sizes and reverse virtual call setup in PCS networks," *Proc. 45th IEEE Vehicular Technology Conference*, vol. 1, pp. 140–144, 1995.

[10] L. Kleinrock, *Queueing Systems, Vol. I: Theory*. John Wiley & Sons, Inc., New York, NY. 1975.

[11] B. Krishnamachari, R. Gau, S. B. Wicker, and Z. J. Hass, "On the performance of sequential paging for mobile user location," Submitted manuscript, 2001.

[12] S. Madhavapeddy, K. Basu, and A. Roberts, "Adaptive paging algorithms for cellular systems," *Proc. 45th IEEE Vehicular Technology Conference*, vol. 2, pp. 976–980, 1995.

[13] U. Madhow, M. L. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 698–707, December 1995.

[14] C. Rose and R. Yates, "Minimizing the average cost of paging under delay constraints," *Wireless Networks*, vol. 1, pp. 211–219, 1995.

[15] C. Rose, "Minimizing the average cost of paging and registration: A timer-based method," *Wireless Networks*, vol. 2, pp. 109–116, 1996.

[16] C. Rose, "State-based paging/registration:a greedy technique," *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 166–173, January 1999.

[17] W. Wang, I. F. Akyildiz, and G. L. Stuber, "Effective paging schemes with delay bounds as QoS constraints in wireless systems," *Wireless Networks*, vol. 7, pp. 455–466, 2001.