Active tracking: Locating mobile users in personal communication service networks

Hanoch Levy and Zohar Naor

Department of Computer Science, School of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel

The problem of tracking mobile users in Personal Communication Service (PCS) networks is discussed. We propose a novel approach for reducing the *wireless* cost of tracking users. The basic idea is to use *non-utilized system resources* for initiating queries about the location of mobile users, in addition to the process of user registration. Queries are applied at each cell, independently of the other cells, whenever the load on the local control channel drops below a pre-defined threshold. Our study focuses on two issues: (1) proposing the initiated queries approach and an algorithm for its application, and (2) studying and quantifying the value of location information and evaluating the parameters affecting it. Our analysis shows that the expected benefit due to location knowledge in a Markovian motion model depends, among other things, on the determinant of the transition matrix and on the variability of the location distribution function. The active tracking approach, as opposed to other dynamic strategies, does not require any modification of user equipment. The importance of this property is in its practicality: An implementation of a new registration strategy in current systems would require a modification of the users equipment. Moreover, the proposed method can be easily implemented *in addition* to any known tracking strategy, to reduce further the tracking cost. The performance of the active tracking method is evaluated under two registration strategies: The geographic-based strategy, currently used in cellular networks, and the profile-based strategy, suggested elsewhere. Under both strategies, it significantly reduces the tracking cost.

1. Introduction

Wireless communication technology is rapidly expanding. Future wireless communication networks will have to support large population of users and will have to provide efficient and low cost services. Location management is one of the key issues in wireless networks: In order to correctly route an incoming call, the cell in which the user resides must be known to the system. Tracking mobile users is based on two elementary operations: *update*, also known as *registration*, in which the user informs the network about its current location, and *find*, also known as *paging*, in which the network searches for the user location. The latter is performed when a connection to the user (for an incoming call) is requested. There is a clear trade-off between the allocation of system resources to paging versus their allocation to registration. Due to this trade-off the proper design and performance issue is that of designing a strategy which minimizes the *combined cost* of paging and registration.

The problem addressed in this work is that of reducing the wireless cost of user tracking, *without any modification of the user equipment*.

Several strategies have been proposed in the literature for tracking users, many of which attempt to reduce the wireless cost associated with user tracking in PCS networks [2,3,5,9]. In [2] all users transmit a *registration* message only at specific cells. A subset of all cells is selected, and designated as reporting cells. Each user will register only upon entering a reporting cell. Thus, when there is a paging event, the search for any user will be restricted to the vicinity of the last reporting cell in which the user has registered. In [9] a *timer based* method was suggested in which the user updates its location every T units of time, where T is a parameter. Each time the user makes no contact with the network for T units of time – it initiates a registration message. Another strategy, suggested in [3,5], is to use *distance based* method in which the user tracks the distance it moved (in terms of cells) from its last known location, and whenever this distance exceeds a parameter D it transmits a registration message.

The basic idea shared by those papers is to use a "partial" user registration: The user does not have to inform the system about each and every location change. Thus, there remains some (bounded) degree of uncertainty about the exact location of the user. Clearly, these schemes must be implemented on the user equipment.

Existing cellular systems use the following tracking strategy, known as the geographic-based strategy: The geographic area is partitioned into *location areas*, based on the commercial licenses granted to the operating companies. A location area (LA) is a group of cells, referred to as a home-system. Users register whenever they change LA, while within the LA the users never register. Thus, when there is an incoming call directed to a user, all the cells within its current LA are paged.

In this work we propose a new approach for minimizing the wireless cost of user tracking. This approach is based on two novel ideas:

1. Use *non-utilized system resources* to obtain information and improve the system knowledge about user location. This idea is based on the fact that due to traffic variability and maximum-capacity network engineering, there is a significant amount of time at which the

network resources are *underutilized*. At such periods these resources can be used, *at practically no cost*, to gather information and improve the network knowledge about user location. For example, in GSM systems the users register only at the borders between location areas. Thus, very often there are non-utilized system resources *within* the location areas.

2. Shift significant parts of the tracking algorithm from the user equipment to the network equipment. The importance of this idea is in its practicality: Since user equipment is distributed among millions of users, no sophisticated algorithm is indeed feasible if it is to be installed on user equipment.

An important building block of our approach is a new unique operation, called an *initiated query*, or a *query*. A query, in nature, is similar to *paging*: A message is sent by a cell requesting a specific user to respond. However, it differs from *paging* in its timing and objective: While *paging* is conducted when a call arrives to the user and its objective is to set up a call, query is conducted at some arbitrary time and its objective is only to increase the knowledge about the user location. Hence, queries are initiated *in addition* to the process of user registration. They are used by the system as a complementary scheme to the registration strategy.

Our study focuses on the evaluation of the efficiency of an initiated query (expressed as the expected reduction in the cost of future paging). To address this issue we propose a general Markovian model for representing user motion. Then, using this model, we derive expressions for the expected reduction in paging cost as a function of the user parameters. These expressions are to be used for two purposes:

(1) Rank the potential queries as to perform the most effective ones, and

(2) Evaluate the overall system performance resulting from using queries. Further we propose approximate expression whose evaluation is of much lower computational complexity. We analyze the expressions and identify several key factors affecting the effectiveness of conducting a query; these are the user call profile, the determinant of the transition matrix, the number of accessible cells, and the variability of the user location probability distribution function. We use numerical results to demonstrate the dependence of the query effectiveness on the user parameters. Simulation results are used to examine the quality of the proposed approximation, and they demonstrate a very good accuracy.

The structure of this paper is as follows: The active tracking strategy is described in section 2. Model and notation are provided in section 3. In section 4 we analyze the paging cost reduction resulting from conducting a query, under the geographic-based registration strategy, and implement the proposed method on a two dimensional random walk model. In section 5 we conduct a similar analysis under the profile-based registration strategy, described in [11]. In section 6 we provide several numerical examples demonstrating the effect of the query operation. Section 7 is devoted to discussion.

2. The active tracking strategy

The tracking strategy is based on using periods at which system resources are underutilized to conduct *initiated queries*, in addition to the process of user registration. This is performed on a cell by cell basis, namely; each cell conducts initiated queries at the specific epochs at which the traffic on its down link control channel (DCCH in GSM system) drops below a pre-defined threshold. Each query is a message, sent by the system, requesting a specific user to respond.

The goal of the queries is to complement the registration process, not to replace it. The importance of this strategy is in the *combination* of centralized and distributed schemes.

Conceptually, periods of low traffic can be used either to increase the registration activity, as it is done in [7], or by a (system initiated) search for the location of those users who are most frequently paged. The last method can be easily implemented, since it requires no modification of user equipment. The method of increasing the registration activity at lowly loaded periods, though it is more efficient, requires a modification of user equipment and user-network interface.

The motivation for the load-sensitive approach is based on the following observations:

(1) There is a significant amount of time, even at working hours, at which the control channel is underutilized for most of the cells within the same LA. The available system resources can be used to reduce the paging cost at other locations and time periods.

(2) Since user equipment is kept as simple as possible, the registration schemes implemented in practice are far from optimal. This situation is unlikely to change in the near future, since an accurate and efficient estimation of call and mobility parameters is still an open problem.

(3) A modification of the registration strategy in current systems would require a modification of the user equipment.

(4) Very often there is a relatively small group of users who consume a significant part of the system's resources allocated for paging. Querying these users is a feasible task which may reduce the signaling load of users tracking.

To illustrate the implementation of the proposed method consider, for example, the timer-based method as adopted by the IS-41 standard: *All* users use the same timer T, regardless their dynamic behavior. The value of T is chosen such that it is reasonable for most of the users. Clearly, the users who are highly mobile, and most frequently paged, need to update their location more frequently, as pointed out in [9]. Since these conditions form a small portion of the users (which consume a significant part of the network resources), this group of users, which may change from the morning to the evening, can be queried more often, such that their effective timer is significantly less than T. The selection of these users can be based on recorded past information, to choose the users with the highest cost of paging, or on the users willingness to pay for a better service.

3. Model and notation

We consider a cellular wireless network consisting of fixed stations, known as base stations (BS), and mobile users. The whole geographic area is partitioned into cells. Each cell is covered by a base station which is connected to the fixed network by wires, and provides a wireless communication link between the mobile users and the network. The user location is understood as an identifier of a cell in which the user is currently residing. We assume that the user location is known to the network just after a *registration*, or a successful *paging* event, and during a conversation.

To model user movement in the network we assume that time is slotted, and that a user can make at most one cell transition during a slot. It is assumed that the movement of the user is done just at the beginning of time slot, such that it precedes any other event, such as a query or a *paging* event. The movements are assumed to be stochastic and independent from one user to another. We assume that queries are performed at the beginning of slots, and they precede any paging event occurring at the same time slot.

The user's *roaming interval* is defined as in [9] and is the time duration between its last contact with the system and the next paging event.

3.1. Motion model

Our motion model is based on a discrete time Markovian model. During each time slot a user can be at any one of the cells in the network. For each user we associate a transition matrix P, whose dimension is $N \times N$ and in which the element P_{ij} is the probability to move from cell i to cell j during one time slot. We assume that the matrix P is ergodic and regular, therefore $\det(P) \neq 0$. The *t-step* transition probability, denoted by p_{ij}^t, is the probability to move from cell i to cell j in t time units, namely, $p_{ij}^t = \Pr[L(t'+t) = j \mid L(t') = i]$, where $L(t)$ denotes the user location at time t, and $p_{ij}^0 = 1$ if $i = j$ and 0 if $i \neq j$. Note that due to the use of Markovian model p_{ij}^t does not depend on t'. Let $\pi_k(t)$ be the probability to be at location k at time t, $t = 0, 1, 2, \ldots$, and let $\pi(t) = (\pi_1(t), \ldots, \pi_N(t))$. The limiting state probability in steady state, $\pi(\infty)$, is denoted by π. The state probability $\pi(t)$ is given by $\pi(t) = \pi(0)P^t$.

The Markovian motion model provides a general framework, which enables the quantification of the value of location information as a function of various parameters, such as user mobility and location distribution. A simpler motion model, which is easier to apply in practice, such as the random walk model, can be easily obtained as a special case of a Markov chain. An implementation of the proposed method on a two dimensional random walk model is described in section 4.4 of this paper.

4. Querying under the geographic-based strategy

In this section we evaluate the performance of an initiated query under the geographic-based (GB) strategy. This is the most common tracking strategy being used in commercial systems. Using the GB strategy, a user registers only at the boundary of each LA. Thus, within the LA there is no registration activity, and the non-utilized system resources can be used for initiated queries.

The first problem addressed is that of ranking the potential queries as to perform the most effective ones. In the next section we derive expressions for the expected reduction in paging cost, resulting from a query. These expressions depend on the steady state probabilities π_i, yielding a memory complexity of $O(N)$, where N is the number of cells within the LA. We therefore propose, in addition, another ranking method, suggested in section 4.2, which is less accurate but whose implementation is much simpler.

4.1. Exact location information (ELI) ranking of queries

The goal of this section is to obtain the expected benefit of a single query, conducted in a single cell. Our first analysis is based on assuming that the user steady state vector π is kept by the system. Thus it can be used for query ranking.

4.1.1. Problem formulation

In this section we evaluate the effect of a query in a single cell. This evaluation is done assuming that the network uses the following paging algorithm: Upon arrival of an incoming call to a user, the network will first search for the user in the location it was last known to be present. If this search fails, the user is then paged simultaneously in all the other cells within the LA. To conduct the analysis we assume that at certain epoch, $t_q > 0$, the system has non-utilized resources in cell l. These resources can be used to initiate queries about the location of some of the users, to increase the system knowledge, and thus to reduce future paging cost.

Our interest is in evaluating the future paging cost reduction due to such a query as a function of the user parameters. This will allow one to rank the possible queries as to perform the most effective ones, and to evaluate the overall system benefit due to queries. Consider a user U whose location was last known to the network at $t = 0$.

Let C be a random variable denoting the cost of paging U, measured as the number of locations needed to be searched to locate U. We are interested in computing the expected value of C, $E[C]$. We assume that outgoing calls are generated by the user as a Poisson process at rate λ_{out} and that incoming calls directed to the user form a Poisson process at rate λ_{in}; let $\lambda = \lambda_{\text{in}} + \lambda_{\text{out}}$. It is assumed that the size of a time slot is sufficiently small such that we can neglect the probability of the arrival of more than one call during a single time slot. The probability that a *roaming interval* will be terminated by a paging event at time t ($t = 0, 1, \ldots$), is the product of: (a) the probability of having no call events during the next t time slots, and (b) the probability of a paging event at time t:

$$p(t) = \left(1 - \mathrm{e}^{-\lambda_{\text{in}}}\right) \mathrm{e}^{-\lambda t}. \tag{1}$$

4.1.2. Paging cost reduction due to query: Exact analysis

Let x be the last known location of U, at time $t = 0$. Let $t_q > 0$ be the time at which U is queried, and $t_p + t_q$ be the next time at which U is paged. Clearly, a query is effective provided that no outgoing call (location update) precedes the next paging event. For this reason our interest is in the case where no *update* event precedes the paging event. Hence, the length of the *roaming interval* is given by $t_q + t_p$, where t_q is the age of the location information refreshed by the query, and t_p is the age of the location information obtained by the query. The interval $(0, t_q)$ is defined as the *past roaming interval*, and the interval $[t_q, t_q + t_p]$ is defined as the *residual roaming interval*.

Let l be the location at which U is queried at t_q, and let $L(t)$ be a random variable denoting the location of U at time t.

To evaluate the cost reduction due to query we distinguish two cases: (1) $L(t_q) \neq l$ and (2) $L(t_q) = l$. In the first case the query performed at t_q yields no response, thus the paging algorithm performed at $t_q + t_p$ will behave the same regardless if a query was performed at t_q or not, yielding identical paging costs under both alternatives.

In the second case, if a query is not conducted at t_q, the paging algorithm conducted at $t_p + t_q$ will search for the user first at x and then in all other locations (assuming that no outgoing call was initiated by the user in the interval $[t_q, t_q + t_p]$). The number of searches required to locate the user is therefore one if $L(t_q + t_p) = x$, or N, if $L(t_q + t_p) \neq x$. The former occurs with probability $p_{lx}^{t_p}$, and the later with probability $(1 - p_{lx}^{t_p})$. The associated cost is therefore:

$$E\left[C \mid L(0) = x, L(t_q) = l, \text{no query}\right] = \left(1 - \mathrm{e}^{-\lambda_{\text{in}}}\right) \sum_{t_p=0}^{\infty} \mathrm{e}^{-\lambda t_p} \left(\left(1 - p_{lx}^{t_p}\right) N + p_{lx}^{t_p}\right). \tag{2}$$

Note that the expression is independent of t_q, since our motion model is memoryless.

On the other hand, if a query is conducted at t_q, it finds the user at location l; thus the paging algorithm conducted at $t_p + t_q$ will search for the user first in l and then in all other locations, yielding

$$E\left[C \mid L(0) = x, L(t_q) = l, \text{query}\right] = \left(1 - \mathrm{e}^{-\lambda_{\text{in}}}\right) \sum_{t_p=0}^{\infty} \mathrm{e}^{-\lambda t_p} \left(\left(1 - p_{ll}^{t_p}\right) N + p_{ll}^{t_p}\right). \tag{3}$$

Let $R_l(t_q \mid L(0) = x)$ be the paging cost reduction when we query for U at t_q at location l. Using that $\Pr[L(t_q) = l \mid L(0) = x] = p_{xl}^{t_q}$, we subtract equation (3) from equation (2) to obtain

$$R_l\left(t_q \mid L(0) = x\right) = (N - 1)\left(1 - \mathrm{e}^{-\lambda_{\text{in}}}\right) p_{xl}^{t_q} \sum_{t_p=0}^{\infty} \mathrm{e}^{-\lambda t_p} \left(p_{ll}^{t_p} - p_{lx}^{t_p}\right). \tag{4}$$

Using that $p_{ll}^{t_p} = e_l P^{t_p} e_l'$, where e_l denotes the row vector whose lth element is 1, and all the others are zero, and e_l' denotes the column vector that is the transpose of e_l, a summation over t_p yields

$$R_l\left(t_q \mid L(0) = x\right) = (N - 1)\left(1 - \mathrm{e}^{-\lambda_{\text{in}}}\right) e_x P^{t_q} \left[I - \mathrm{e}^{-\lambda} P\right]^{-1} \left[e_l' - e_x'\right]. \tag{5}$$

Note that $R_l(t_q \mid L(0) = x)$ may get negative values, namely, a query may increase the expected paging cost. This is since the paging algorithm ignores the vector $\pi(t)$.

4.1.3. Approximation

Due to the relatively high computational complexity of equation (5), an attempt to implement it on a realistic situation (when N is large), may not be feasible. For this reason we seek an approximate expression for the expected cost reduction, whose computational complexity is much lower.

Using the *Shrinkage Factor* [4], we suggest to approximate the difference between $\pi_j(t)$ and $\pi_j(\infty)$ by

$$\left|\pi_j(\infty) - \pi_j(t)\right| \approx K \left|\det(P)\right|^{t/(N-1)}, \tag{6}$$

where K is a constant to be determined. We choose K such that the approximation for $\pi_j(t)$ is exact for $t = 0$ and for $t = \infty$. Thus,

$$\pi_j(t) \approx \pi_j + \left[\pi_j(0) - \pi_j\right] \left|\det(P)\right|^{t/(N-1)}, \tag{7}$$

where, from now on, we denote $\pi(\infty)$ by π and $\pi_j(\infty)$ by π_j.

Let us define the *stationarity factor* S by $S = |\det(P)|^{1/(N-1)}$. Since $S^{-(N-1)}$ reflects the (geometric) rate at which the Markov chain converges to its steady state, we identify S ($0 < S < 1$) as the user tendency to remain in its current location. The *mobility factor* is defined as S^{-1}. When S approaches 1, P approaches the unity matrix, the user location is almost fixed over time, and the *mobility factor* is minimal. When S approaches 0, the user reaches steady state instantaneously, regardless of its initial state, therefore the user mobility is maximized.

To find an upper bound on the approximation suggested in equation (7), let $\Lambda_{\max}$ be the second largest eigenvalue of the matrix P. Since P is ergodic, $|\Lambda_{\max}| < 1$ [6]. Sinclair and Jerrum [10], in continuation to previous work by Alon [1] on eigenvalues and expanders graphs, obtained an upper bound on the distance between $\pi_j(t)$ and π_j:

$$\left|\pi_j(t) - \pi_j\right| \leqslant \frac{\pi_j}{\pi_{\min}}\Lambda^t_{\max}, \tag{8}$$

where $\pi_{\min}$ is the smallest component of π: $\pi_{\min} = \min\{\pi_i\colon 1 \leqslant i \leqslant N\}$. Let $\pi'_j(t)$ be the approximated value of $\pi_j(t)$ given in equation (7). Then,

$$\left|\pi'_j(t) - \pi_j\right| = \left|\pi_j(0) - \pi_j\right| S^t. \tag{9}$$

Let $\varepsilon(t)$ be the distance between the approximated and real value of $\pi_j(t)$. Using the triangle inequality in equations (8) and (9) we get

$$\varepsilon(t) = \left|\pi'_j(t) - \pi_j(t)\right| \leqslant \frac{\pi_j}{\pi_{\min}}\Lambda^t_{\max} + \left|\pi'_j(t) - \pi_j\right|. \tag{10}$$

Substitute equations (8), (9) in equation (10) yields

$$\varepsilon(t) \leqslant \begin{cases} \dfrac{\pi_j}{\pi_{\min}}\Lambda^t_{\max} + \pi_j S^t & \text{if } \pi_j \geqslant 1/2, \\ \dfrac{\pi_j}{\pi_{\min}}\Lambda^t_{\max} + (1-\pi_j) S^t & \text{if } \pi_j < 1/2. \end{cases} \tag{11}$$

Thus, the upper bound on the error of the approximation given in equation (7) is negligible, provided that $\pi_{\min}$ is not extremely small relative to other π_i's, and that $\Lambda_{\max}$ is bounded away from 1. Both conditions hold for a broad class of Markov chains (e.g., random walk), which are suitable for motion modeling in PCS networks, and are rarely violated in practice. Nevertheless, to examine the approximation accuracy, we used simulation on Markov chains which some of them contain states satisfying the condition: $\{\exists j\colon \pi_j \gg \pi_{\min}\}$. The average error over all states and vectors was negligible. It should be noted that in all the examples that were used in the simulation, the approximation error was much lower than the upper bound described in equation (11). The reason is that the upper bound corresponds to situations where either $\pi_j(t) < \pi_j < \pi'_j(t)$ or $\pi_j(t) > \pi_j > \pi'_j(t)$. In reality, these situations emerge only for large t, when $\varepsilon(t) \approx 0$. When t is small, we found that in practice

$$\varepsilon(t) < \left|\frac{\pi_j}{\pi_{\min}}\Lambda^t_{\max} - S^t\right|.$$

The simulation results are described in section 6, figure 4.

Using equation (7) repeatedly, where $\pi_l(t_q) = 1$, $\pi_x(t_q) = 0$, we obtain

$$p^{t_p}_{ll} - p^{t_p}_{lx} \approx \pi_l - \pi_x + \left[1 - (\pi_l - \pi_x)\right] S^{t_p} \quad (l \neq x). \tag{12}$$

Thus, equation (4) yields

$$R_l\big(t_q \mid L(0) = x\big) \approx (N-1)\big(1 - \mathrm{e}^{-\lambda_{\text{in}}}\big)\pi_l(t_q) \times \sum_{t_p=0}^{\infty} \mathrm{e}^{-\lambda t_p}\big((\pi_l - \pi_x) + \left[1 - (\pi_l - \pi_x)\right] S^{t_p}\big). \tag{13}$$

If $l \neq x$ then $L(0) = x$ implies $\pi_l(t_q) \approx (1 - S^{t_q})\pi_l$, which finally yields

$$R_l\big(t_q \mid L(0) = x\big) \approx \begin{cases} (N-1)\big(1 - \mathrm{e}^{-\lambda_{\text{in}}}\big)\big(1 - S^{t_q}\big) \\ \quad \times \pi_l \left[\dfrac{\pi_l - \pi_x}{1 - \mathrm{e}^{-\lambda}} + \dfrac{1 - (\pi_l - \pi_x)}{1 - \mathrm{e}^{-\lambda} S}\right] & (l \neq x), \\ 0 \quad (l = x), \end{cases} \tag{14}$$

where the equation for $l = x$ follows from substituting $p^{t_p}_{ll} - p^{t_p}_{lx} = 0$ in equation (4). Equation (14) obtains the paging cost reduction on the basis of exact location information (ELI), i.e., the last known location of the user, and the inspected location. Hence, it can be used to rank the potential queries as to perform the most effective ones. Note that equation (14) implies that a query in a single cell may, in some situations, increase the expected paging cost; this is in agreement with equation (5). The condition under which $R_l(t_q \mid L(0) = x) < 0$ is given by

$$\pi_x - \pi_l > \frac{\mathrm{e}^{\lambda} - 1}{1 - S}. \tag{15}$$

Using that $1 > \pi_x - \pi_l$ we get that a sufficient condition under which $R_l(t_q \mid L(0) = x) > 0$ is

$$\lambda \geqslant \ln(2 - S). \tag{16}$$

Thus, for a user with high call rate, or high *stationarity factor* (i.e., low mobility), a query will always yield a positive gain. The reason for the condition specified in equation (15) is that the paging algorithm is not suitable for highly mobile users (low S), with low call rate (low λ). As a result, for a user with high location variability, i.e., $\pi_x \gg \pi_l$, a query may yield a negative gain.

Equation (14) implies that the potential benefit of a query is mainly affected by the following factors:

- The value of prior knowledge S^{t_q}, where t_q is the age of prior location information (i.e., the length of the *past roaming interval*). The term $(1 - S^{t_q})$ reflects the value of refreshing the location information; for example, if $t_q = 0$ this value is zero. Hence, S reflects the *aging rate* at which the location information dissolves as a function of time. Since S^{t_q} reflects the value of prior location information, the reduction in paging cost increases with the age of location information t_q. This observation is the basis for the timer-based method suggested in [9].
- The rates of paging events and update messages. The query gain increases with λ_{in} and decreases with λ_{out}. This results from the fact that a query is effective in situations where the next incoming call precedes the next location update.
- The future value of the location information obtained by the query, S^{t_p}, where t_p is the length of the *residual roaming interval*. This value increases with S, reflecting the property that under smaller convergence rate the future value of location information is higher. Note the

trade-off between the benefit of the future value obtained by a query, and the benefit of refreshing prior location knowledge. The first increases with S (through the term $1/(1 - e^{-\lambda}S)$) while the later decreases with S, through $(1 - S^{t_q})$.

- The difference between the user tendency to reside in the inspected location l and its tendency to reside in the last known location x, expressed by $\pi_l - \pi_x$. The higher this difference, the most effective the query. This factor implies that a query may yield a negative gain, especially for situations where $\pi_x \gg \pi_l$. It should be noted that the factor $\pi_x - \pi_l$ comes from the probability that the user will return to its last known location x before the next paging event. This probability also depends on S and λ (see equation (15)). Since the gain resulting from a query in cell l increases with the likelihood π_l, if a single user is to be queried in many cells, the search (query) should be conducted in decreasing order of the probabilities π_l. This is in agreement with the result obtained by Rose and Yates [8], that optimal paging is achieved by paging the cells in decreasing order of the user likelihood to be in them.

Equation (14) forms a mechanism for query ranking, based on exact location information, namely, the last known location and the inspected location. Query ranking using Exact Location Information (ELI) can be used to eliminate unnecessary queries, whose expected gain is non-positive. However, its implementation requires to maintain the steady state vector π for each user, yielding a memory complexity of $O(N)$ *for each user*.

4.2. Query ranking assuming Homogeneous Location Distribution (HLD)

In this section we evaluate the expected benefit of refreshing the knowledge about the *exact* user location. Assuming homogeneous location distribution (HLD), this value can be used to estimate the potential benefit of a single query. The ELI ranking, presented in section 4.1 (equation (14)), requires to keep the steady state vector π for each user. Below we propose an alternative ranking, which is less accurate, but whose complexity is lower, since it does not require to keep the vector π. The basic idea is to rank the various queries using only general user parameters, assuming that in steady state, the user location distribution is homogeneous. Query ranking assuming homogeneous location distribution (HLD) is uniform in the sense that a user is given the same rank in all cells, as opposed to ELI ranking.

As a first step, we evaluate the expected paging cost reduction for a specific user, assuming that a query is conducted all over the network. Let $R(t_q)$ denote the total paging cost reduction under this operation, namely: $R(t_q) = \sum_{l=1}^{N} R_l(t_q)$. Given that the last known location of the user is x, the expected reduction in the cost of *paging* is obtained using equation (5):

$$\begin{aligned} R(t_q \mid L(0)=x) &= \sum_{l=1}^{N} R_l(t_q \mid L(0)=x) \\ &= (N-1)(1-e^{-\lambda_{\text{in}}}) e_x P^{t_q} \\ &\quad \times [I - e^{-\lambda}P]^{-1} (\mathcal{I} - N e'_x), \end{aligned} \quad (17)$$

where $\mathcal{I}$ is the unit column vector all of whose elements are 1.

Using the same approximation described in section 4.1, we obtain from equation (14)

$$\begin{aligned} &R(t_q \mid L(0)=x) \\ &\quad \approx (N-1)(1-e^{-\lambda_{\text{in}}})(1-S^{t_q}) \\ &\qquad \times \left[\frac{\sum_{i=1}^{N} {\pi_i}^2 - \pi_x}{1-e^{-\lambda}} + \frac{1-(\sum_{i=1}^{N} {\pi_i}^2 - \pi_x)}{1-e^{-\lambda}S}\right]. \end{aligned} \quad (18)$$

We define the *variability factor* of the steady state vector π as

$$\nu = \sum_{i=1}^{N} {\pi_i}^2. \quad (19)$$

The certainty about the user location increases with ν, where $1/N \leqslant \nu < 1$; the minimum value of ν is achieved for homogeneous location distribution, reflecting total uncertainty about the user location in steady state. Using the parameter ν we obtain

$$\begin{aligned} R(t_q \mid L(0)=x) &\approx (N-1)(1-e^{-\lambda_{\text{in}}})(1-S^{t_q}) \\ &\quad \times \left[\frac{\nu-\pi_x}{1-e^{-\lambda}} + \frac{1-(\nu-\pi_x)}{1-e^{-\lambda}S}\right]. \end{aligned} \quad (20)$$

It should be noted that $R(t_q \mid L(0)=x)$ may take negative values, when $\pi_x \gg \nu$. Assuming unconditional queries (namely, even for non-positive gain), the expected paging cost reduction is

$$\begin{aligned} R(t_q) &= \sum_{x=1}^{N} \pi_x R(t_q \mid L(0)=x) \\ &\approx \frac{(N-1)(1-e^{-\lambda_{\text{in}}})(1-S^{t_q})}{1-e^{-\lambda}S}. \end{aligned} \quad (21)$$

Remark 1. Note that the expected paging cost reduction resulting from the *exact* knowledge about the user location at time t_q depends only on the *stationarity factor* S, the *past roaming interval* t_q, and on λ_{in}, λ, and not on ν.

Assuming homogeneous location distribution (HLD), the expected paging cost reduction caused by a single query (i.e., in a single cell) is

$$R'(t_q) \approx \frac{(N-1)(1-e^{-\lambda_{\text{in}}})(1-S^{t_q})}{N(1-e^{-\lambda}S)}. \quad (22)$$

Equation (22) may serve as a basis for ranking the various queries using only general user parameters. Being less accurate, its performance should be inferior to the ELI ranking. However, the complexity of its implementation is significantly lower.

Remark 2. Note that substitution of $\forall l\ \pi_l = N^{-1}$ in equation (14) yields

$$R_l\big(t_q \mid L(0) = x\big) = R'(t_q), \quad \forall l, x.$$

In other words: under homogeneous location distribution ELI ranking yields the same rank as HLD ranking, as expected.

4.3. Comparison of query ranking criteria

Two methods for ranking the various queries were proposed:

(1) ELI, based on the benefit resulting from applying a single query in a specified location (equation (14)), and

(2) HLD, based on the assumption that, in steady state, the user location distribution is homogeneous (equation (22)).

The ELI ranking depends on the last known location of the user and the inspected location. Thus, the algorithm requires to hold the steady state vector π, yielding a memory complexity of $O(N)$ per user. HLD, in contrast, may be applied directly, ignoring exact location information. Assuming fixed calling rates and mobility pattern, the implementation of HLD requires to keep for each user only four static parameters: N, λ_{out}, λ_{in}, S, and one dynamic parameter: t_q. Therefore, its memory complexity is $O(1)$. It should be noted that using HLD is similar to the timer-based registration method [9].

In general, ELI ranking is more accurate than HLD. However, in some situations, the detailed location information used by ELI, has only a marginal contribution to the query effectiveness. In these situations, both methods should give approximately the same rank. The purpose of this section is to quantify the difference in the effectiveness of the ranking methods, and to isolate these situations, in which HLD ranking can replace ELI.

To compare the performance of these methods let us define δ as the difference in the rank given by them. Using equations (14) and (22) we obtain

$$\begin{aligned}\delta(l,x) &= \big|R_l\big(t_q \mid L(0) = x\big) - R'(t_q)\big| \\ &\approx R(t_q)\left|\left(1 + \frac{(\pi_l - \pi_x)(1-S)}{e^{\lambda} - 1}\right)\pi_l - \frac{1}{N}\right|. \end{aligned} \tag{23}$$

Let $\theta(x)$ be the difference between the ELI ranking and the HLD ranking, given that $L(0) = x$, and assuming that queries are conducted *simultaneously* all over the network. Then:

$$\begin{aligned}\theta(x) &= \sum_l \delta(l,x) \\ &< R(t_q)\left[\sum_l \left(\left|\pi_l - \frac{1}{N}\right| + \left|\frac{(1-S){\pi_l}^2}{e^{\lambda} - 1}\right|\right) \right.\\ &\qquad \left. + \left|\frac{(1-S)\pi_x\pi_l}{e^{\lambda} - 1}\right|\right]. \end{aligned} \tag{24}$$

Let Φ be the *accumulated deviation from the average* of the steady state location probability: $\Phi = \sum_l |\pi_l - 1/N|$. Note that $0 \leqslant \Phi < 2 - 2/N$. Substitute Φ in equation (24) we obtain

$$\theta(x) < R(t_q)\left[\Phi + \frac{(1-S)\nu}{e^{\lambda} - 1} + \frac{(1-S)\pi_x}{e^{\lambda} - 1}\right]. \tag{25}$$

The expected total difference, per user, between ELI ranking and HLD ranking is therefore

$$\eta = \sum_x \pi_x \theta(x) < R(t_q)\left[\Phi + \frac{2(1-S)\nu}{e^{\lambda} - 1}\right]. \tag{26}$$

Thus, the relative difference is

$$\Delta = \frac{\eta}{R(t_q)} < \Phi + \frac{2(1-S)\nu}{e^{\lambda} - 1}. \tag{27}$$

The upper bounds on η and Δ monotonically increase with the location variability, reflected by ν and Φ, where for homogeneous location distribution we get $\eta = 0$ (remark 2). Hence, for relatively small location variability HLD may replace ELI, since there is no need for exact location information. On the other hand, the difference in the efficiency of ELI and HLD becomes significant for highly mobile users ($S \approx 0$), with low call rate ($\lambda \approx 0$) and high location variability. Given the steady state vector π for each user, the parameters η and Δ can be used to adapt the query ranking method to the user parameters.

4.4. A demonstration on a random walk model

In this section the proposed method is implemented on a two dimensional random walk model with a grid topology. The expected reduction in paging cost is evaluated using an exact analysis. Then the results are compared to the results obtained by the approximation method provided in equation (21).

Consider a system with N^2 cells, arranged in a two dimensional array whose dimension is $N \times N$. A user located in cell i can remain in i with probability p, or move to any one of its nearest neighbors, in horizontal, vertical or diagonal direction, with probability $(1-p)/n_i$, where n_i is the number of nearest neighbors of the cell i. The transition matrix P, whose dimension is $N^2 \times N^2$, is given by

$$P_{ij} = \begin{cases} p, & \text{if } i = j, \\ \dfrac{1-p}{n_i}, & \text{if } i \text{ is a nearest neighbor of } j, \\ 0, & \text{if } i \text{ is not a nearest neighbor of } j. \end{cases} \tag{28}$$

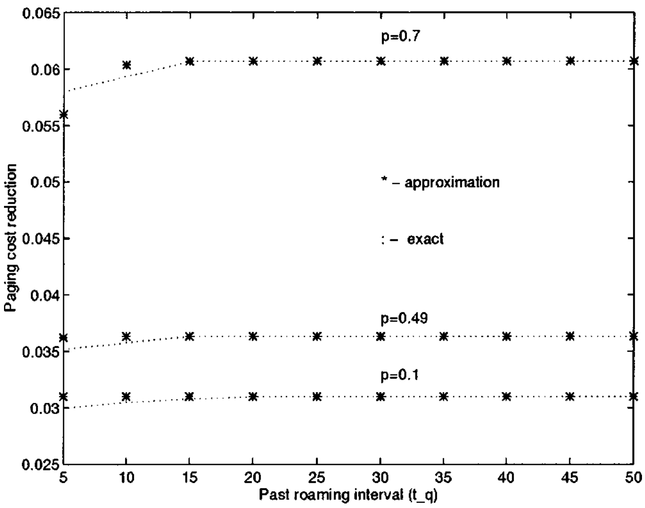

Figure 1. Paging cost reduction – using an exact and approximation analysis.

Since the location distribution under this motion model is approximately homogeneous, the expected benefit of the active tracking strategy is given by equation (21), while the benefit of a single query is given in equation (22. Thus, all the required information on user behavior is the call profile (λ_{in}, λ_{out}), the past roaming interval, t_q, and the *stationarity factor* S. Since S can be derived from the probability p, which determines the value of $|\det(P)|$, *there is no need to hold a transition matrix for each user* – a single number (S) is sufficient. This property is unique for the homogeneous location distribution. Moreover, since the steady state location probability satisfies: $\pi_j/\pi_{\min} \approx 1$, $\forall j$, the upper bound on the approximation error, given in equation (11), is negligible. This implies that the approximation given in equation (21) should be close enough to the exact result.

To demonstrate our method, we apply the active tracking strategy on a system with 4 cells, arranged in an array whose dimension is 2×2, which forms a clique. The transition matrix P whose dimension is 4×4 is given by: $P_{ij} = p$ if $i = j$, and $P_{ij} = (1-p)/3$ if $i \neq j$. The steady state probability is $\pi_i = 0.25$ for $i = 1, 2, 3, 4$. The expected benefit of queries was evaluated using an exact analysis (equation (5)) and the approximation given in equation (21), for $\lambda_{\text{in}} = \lambda_{\text{out}} = 1/120$, and for $p = 0.1$, 0.49, 0.7.

The expected reduction in paging cost, as evaluated by both methods, is depicted in figure 1. The difference between the approximated and exact results is negligible. However, it should be noted that for this specific example, $\pi_j/\pi_{\min} = 1$, which gives a negligible approximation error, especially for large t_q. The expected reduction in paging cost is relatively modest, since the number of cells, 4, is small.

5. Querying under the profile-based strategy

To further study the initiated query approach we evaluate the expected reduction in the paging cost under the profile-based (PB) strategy, suggested in [11]. In that strategy the network maintains a sequential list of the most likely locations in which the user can be found. The list is ordered from the most to the least likely location. When a paging event occurs, the list is searched in a sequential order, one location at a time, from the most to the least likely location. The user does not register while moving between locations in the list.

The PB strategy is the opposite extreme of the geographic-based (GB) strategy, in the sense that while the GB strategy makes no assumptions about the user location and mobility, and the paging delay is minimal, the PB strategy assumes a detailed knowledge about the user mobility pattern, and the paging delay is relatively high. Rose and Yates [8] showed that, considering a single user and ignoring paging delay, an optimal paging is achieved by paging the cells in decreasing order of the user likelihood to be in them. Thus, this paging strategy is an optimal, in the sense that the expected paging cost is minimal.

As in section 4, we assume that a query is conducted at the beginning of a time slot, such that if there is a paging event in the same slot, then the query operation precedes the paging operation. The *paging* algorithm assumes that the network keeps, for each user, at every time slot t, a location probability vector $\pi(t)$, where $\pi_i(t)$ is the probability to find the user at cell i at time t. The *paging* algorithm is based on sequential search for the user in the cells, in decreasing order of probabilities $\pi_i(t)$. Let $\alpha(\pi(t))$ be the vector $\pi(t)$ sorted in non-increasing order. There exists a column vector $\gamma(\pi(t))$, a permutation over $[N]$ such that $\gamma_i(\pi(t))$ is the index of $\pi_i(t)$ in the vector $\alpha(\pi(t))$. Then, given that the length of the *roaming interval* is t, the expected paging cost, in terms of number of locations being searched, is

$$E\bigl[C(t)\bigr] = \sum_{i=1}^{N} i\alpha_i(t) = \pi(t)\gamma\bigl(\pi(t)\bigr). \tag{29}$$

We shall now compute the reduction in the paging cost, due to a query in cell l. As defined earlier, let t_q denote the time of query, $t_q + t_p$ the length of the *roaming interval*, and let $L(0) = x$. Let $k = \gamma_l(\pi(t_q))$. Assuming that the next paging event occurs at $t_p = 0$, the expected paging cost at time $t = t_q$, without query in l, is given by

$$\begin{aligned} &E\bigl[C(t_q) \mid \text{no query}\bigr] \\ &\quad= \sum_{j=1}^{N} j\alpha_j(t_q) \\ &\quad= \sum_{j<k} j\alpha_j(t_q) + k\pi_l(t_q) + \sum_{j>k} j\alpha_j(t_q). \end{aligned} \tag{30}$$

When a query is conducted, the user is found in l with probability $\pi_l(t_q)$, and is not found in l with probability $(1 - \pi_l(t_q))$. If the user is found, $\pi(t_q)$ changes to $\pi^1(t_q)$, where

$$\pi_i^1(t_q) = \begin{cases} 1, & i = l, \\ 0, & i \neq l. \end{cases}$$

If the user is not found at l, $\pi(t_q)$ changes to $\pi^2(t_q)$, where

$$\pi_i^2(t_q) = \begin{cases} 0, & i = l, \\ \dfrac{\pi_i(t_q)}{1-\pi_l(t_q)}, & i \neq l. \end{cases}$$

Thus, the expected immediate cost of paging after conducting a query in l is

$$\begin{aligned} &E\left[C(t_q) \mid \text{query in } l\right] \\ &\quad = \pi_l(t_q)\pi^1(t_q)\gamma\left(\pi^1(t_q)\right) \\ &\qquad + \left(1-\pi_l(t_q)\right)\pi^2(t_q)\gamma\left(\pi^2(t_q)\right). \end{aligned} \tag{31}$$

Substitution of π^1 and π^2 in equation (31) yields

$$\begin{aligned} &E\left[C(t_q) \mid \text{query in } l\right] \\ &\quad = 1\pi_l(t_q) + \sum_{j<k} j\alpha_j(t_q) + \sum_{j>k}(j-1)\alpha_j(t_q). \end{aligned} \tag{32}$$

Let $R_l(t_q)$ be the expected reduction in paging cost when we query in cell l at time t_q:

$$R_l(t_q) = \left(1-\mathrm{e}^{-\lambda_{\text{in}}}\right)\sum_{t_p=0}^{\infty} \mathrm{e}^{-\lambda t_p} R_l(t_q \mid t_p). \tag{33}$$

Subtracting equation (32) from equation (30) yields

$$\begin{aligned} R_l(t_q \mid t_p = 0) &= (k-1)\pi_l(t_q) + \sum_{j>k}\alpha_j(t_q) \\ &= (k-1)\pi_l(t_q) + \left(1-\Pi_l(t_q)\right) \end{aligned} \tag{34}$$

where $\Pi_l(t_q)$ is the cumulative distribution of $\pi_l(t_q)$, defined by

$$\Pi_l(t_q) = \sum_{m=1}^{\gamma_l(\pi(t_q))} \alpha_m(t_q). \tag{35}$$

Conditioning on the value of t_p, $R_l(t_q \mid t_p)$ is given by

$$\begin{aligned} R_l(t_q \mid t_p) &= E\left[C(t_q+t_p) \mid \text{no query}\right] \\ &\quad - E\left[C(t_q+t_p) \mid \text{query in } l\right], \end{aligned}$$

which leads to

$$\begin{aligned} R_l(t_q \mid t_p) &= \pi(t_q+t_p)\gamma\left(\pi(t_q+t_p)\right) \\ &\quad - \pi_l(t_q)\pi^1(t_q+t_p)\gamma\left(\pi^1(t_q+t_p)\right) \\ &\quad - \left(1-\pi_l(t_q)\right)\pi^2(t_q+t_p)\gamma\left(\pi^2(t_q+t_p)\right) \end{aligned} \tag{36}$$

where

$$\pi^1(t_q+t_p) = \pi^1(t_q)P^{t_p}, \tag{37}$$

$$\pi^2(t_q+t_p) = \pi^2(t_q)P^{t_p}. \tag{38}$$

For large N, the exact computation of equation (36) for each and every user is impractical. We, therefore, seek for a simpler, though approximate expression. Substitute $\pi_l^1(t_q) = 1$, $\pi_l^2(t_q) = 0$, we get, using the approximation suggested in equation (7),

$$\pi_i^1(t_q+t_p) \approx \begin{cases} \left(1-S^{t_p}\right)\pi_l + S^{t_p}, & i = l, \\ \left(1-S^{t_p}\right)\pi_i, & i \neq l, \end{cases} \tag{39}$$

$$\pi_i^2(t_q+t_p) \approx \begin{cases} \left(1-S^{t_p}\right)\pi_l, & i = l, \\ \left(1-S^{t_p}\right)\pi_i + \dfrac{\pi_i(t_q)}{1-\pi_l(t_q)}S^{t_p}, & i \neq l. \end{cases} \tag{40}$$

Given that $L(0) = x$ we get

$$\pi_i(t_q+t_p) \approx \begin{cases} \left(1-S^{t_p+t_q}\right)\pi_i, & i \neq x, \\ \left(1-S^{t_p+t_q}\right)\pi_i + S^{t_p+t_q}, & i = x. \end{cases} \tag{41}$$

Note that equations (39)–(41) imply that: $\pi_l^1(t_q+t_p) > \pi_l(t_q+t_p) > \pi_l^2(t_q+t_p)$, and for $i \neq l$: $\pi_i^1(t_q+t_p) < \pi_i(t_q+t_p) < \pi_i^2(t_q+t_p)$. Hence,

$$\gamma\left(\pi_l^1(t_q+t_p)\right) \leqslant \gamma\left(\pi_l(t_q+t_p)\right) \leqslant \gamma\left(\pi_l^2(t_q+t_p)\right),$$

and for $i \neq l$:

$$\gamma\left(\pi_i^1(t_q+t_p)\right) \geqslant \gamma\left(\pi_i(t_q+t_p)\right) \geqslant \gamma\left(\pi_i^2(t_q+t_p)\right).$$

Substitute equations (39)–(41) in equation (36) suggests that the gain results from conducting a query monotonically shrinks in a factor of S^{t_p}, until $\pi^1 \approx \pi^2 \approx \pi$, where we assume that the *rate* at which $\gamma(\pi_i)$ changes in time, is approximately the rate at which π_i changes (clearly, the exact *amount* of change must depend on the values of the other components of π). This assumption is certainly not true for short time periods, since $\gamma(\pi_i)$ is a discrete number. However, for sufficiently long time interval, we can make this assumption, which suggests the following approximation:

$$R_l(t_q \mid t_p) \approx R_l(t_q \mid t_p = 0)S^{t_p} \tag{42}$$

to obtain

$$R_l(t_q) \approx \left(1-\mathrm{e}^{-\lambda_{\text{in}}}\right)R_l(t_q \mid t_p = 0)\sum_{t_p=0}^{\infty}\mathrm{e}^{-\lambda t_p}S^{t_p}, \tag{43}$$

which finally yields

$$R_l(t_q) \approx \frac{(1-\mathrm{e}^{-\lambda_{\text{in}}})R_l(t_q \mid t_p = 0)}{1-\mathrm{e}^{-\lambda}S}. \tag{44}$$

Remark 3. Note the similarity of equation (44) to equation (14): They both reflect the property that under high convergence rate the future value of location information is small. The dependency of the query gain on the user parameters remains the same under both paging algorithms.

6. Numerical results

To demonstrate the effect of the query prioritizing mechanism we examine several numerical examples. We consider an LA consisting of 10 cells and evaluate the expected cost reduction as a function of the *roaming interval* length, and the user mobility characteristics, expressed by the *stationarity factor* S. We consider three users, having the same steady state probability vector π, and differing in their *stationarity factor S* (low, medium and high).

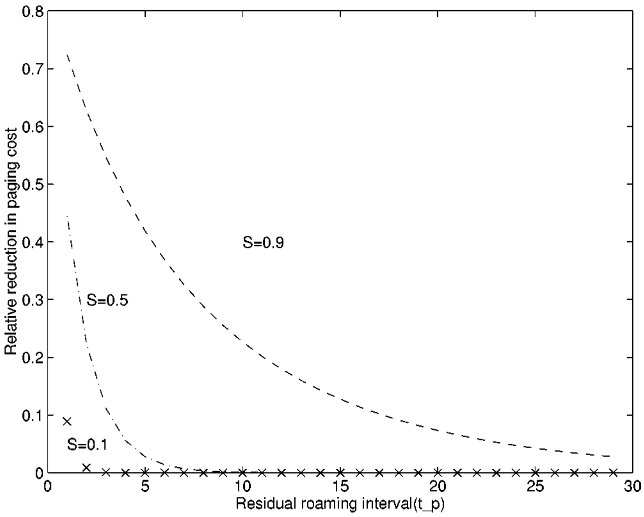

Figure 2. Effects of mobility and residual roaming interval length on paging cost reduction.

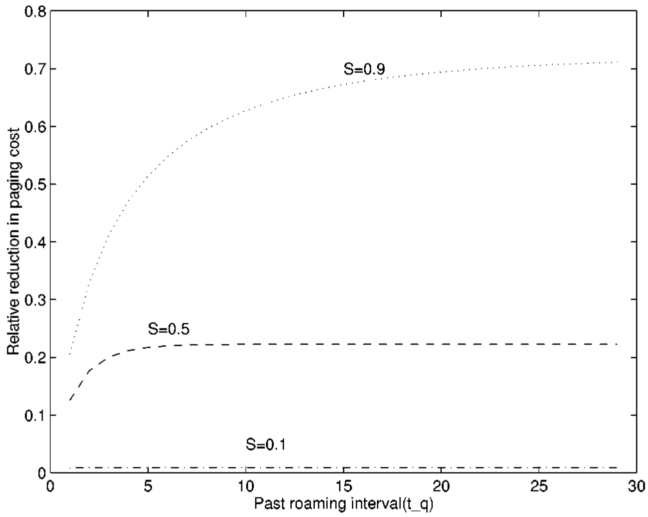

Figure 3. Effects of value of prior location information S^{t_q} on paging cost reduction.

Figure 2 demonstrates the dependency of the query effectiveness on the *residual roaming interval*. Assuming that the length of the *roaming interval* is $t_q + t_p$, and that a query is conducted at time t_q all over the network, the relative reduction in the expected paging cost resulting from this operation is given by

$$\Xi(t_q, t_p) = 1 - \frac{\sum_x \pi_x \sum_l p_{xl}^{t_q} E[C(t_q + t_p) \mid L(t_q) = l]}{\sum_x \pi_x E[C(t_q + t_p) \mid L(0) = x]}, \quad (45)$$

where $E[C(t)]$ is the expected paging cost, in terms of number of locations being searched, defined in section 4.1.1, given that the length of the *roaming interval* is t.

In figure 2 the relative reduction in the expected paging cost, $\Xi(t_q, t_p)$, is plotted as a function of the *residual roaming interval* t_p (i.e., the age of the location information obtained by the query). As expected, the query efficiency increases with S, and decreases with t_p, since the information gained by the query dissolves over time. For example, for $t_p = 1$, a reduction of 72% from the expected paging cost is achieved for $S = 0.9$, in comparison to 9% reduction for $S = 0.1$.

Figure 3 depicts the relative cost reduction $\Xi(t_q, t_p)$ as a function of the length of the *past roaming interval*, t_q, i.e., the age of prior location information refreshed by the query, for the same three user types. Recall that the *stationarity factor* S also reflects the *aging rate*, the *value of prior location knowledge* is reflected by S^{t_q}. The benefit of a query monotonically increases with the age of location information and the *aging rate*, and asymptotically approaches a certain value (which depends on the *aging rate*). The reason for that is that the higher the age of the location information, the larger is the benefit from refreshing it. The rate at which the query gain reaches its maximum value decreases with the *aging rate* (i.e., increases with the *mobility* S^{-1}), and for highly mobile user, the expected gain reaches its maximum very soon. Both figures indicate that highly mobile users should be queried more often, as one would intuitively expect.

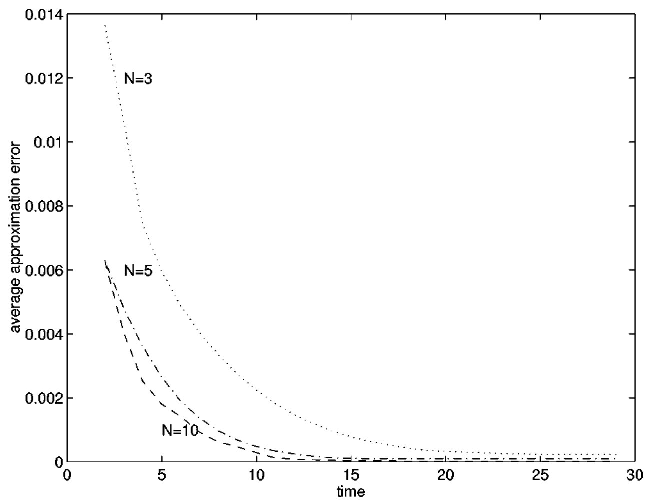

Figure 4. Time dependency of approximation accuracy.

Figure 4 examines the quality of the approximation proposed in equation (7). The examination is conducted by comparing the real value of each component of the vector π to the approximated value of the same component. We consider three matrix sizes, having dimensions of $N = 3$, 5, 10. For each matrix size nine different matrices were examined, representing different combinations of large, medium and small ν and S. The arithmetic average of the approximation errors over all the $(9N)$ cases was plotted as a function of time. The results show that the difference between real and approximated values is negligible, and decreases rapidly with time.

7. Summary

We proposed a new approach for tracking mobile users in wireless networks. We introduced the concept of *initi-*

ated query, an operation similar to paging which is used by the system to refresh its knowledge about the user location. Queries are initiated by the system (as opposed to *registration*) and thus can be activated at times convenient to the system, namely at low utilization epochs. As such, queries are virtually cost free and bear with them benefits only.

We introduced a Markovian motion model that is required in order to evaluate queries in a general framework. We analyzed the cost savings associated with a query as a function of the user parameters. We identified key parameters affecting the effectiveness of a query: User mobility, location variability, age of location information, aging rate of location information, volumes of incoming paging events and location updates; we evaluated the effects of these parameters on query effectiveness.

Our results show, that significant reduction of the paging cost can be achieved if initiated queries are used. Furthermore, they show that the savings are quite sensitive to the user parameters; thus, discrimination between less effective and more effective queries, based on user parameters, can contribute significantly to the query efficiency. Our analysis provided an *exact* criterion (equation (5)) for such discrimination; however, the implementation of this criterion is of high complexity. Therefore, we used an approximation method to evaluate the behavior of a Markov chain (equation (7)), whose implementation is of much lower complexity, and whose performance (see figure 4) is very close to that of the exact one. On the basis of this approximation, we derived a criterion (equation (14)) whose implementation is of much lower complexity. However, it still requires memory complexity of $O(N)$ per user, since the queries are ranked using exact location information (ELI).

We therefore proposed an alternative criterion (equation (22)), by which queries are ranked *only* on the basis of general user parameters, assuming homogeneous location distribution (HLD), which leads to memory complexity of O(1) for a single user. Although ELI ranking is, in general, superior to HLD ranking, in many cases their performance is very close. Our analysis shows that the difference in the efficiency of these ranking methods depends on user parameters: Location variability, mobility, and expected rate of paging and update events. This dependency may be used to adapt the query ranking method to the user parameters.

References

[1] N. Alon, Eigenvalues and expanders, Combinatorica 6 (1986) 83–96.

[2] A. Bar-Noy and I. Kesler, Tracking mobile users in wireless networks, IEEE Transactions on Information Theory 39 (November 1993) 1877–1886.

[3] A. Bar-Noy, I. Kessler and M. Sidi, Mobile users: To update or not to update? Wireless Networks 1(2) (1995) 175–185.

[4] R.A. Howard, *Dynamic Probabilistic Systems*, Vol. 1, *Markov Models* (Wiley, New York, 1971).

[5] U. Madhow, L. Honig and K. Steiglitz, Optimization of wireless resources for personal communications mobility tracking, IEEE Transactions on Networking 3(6) (1995) 698–707.

[6] E. Seneta, *Non-Negative Matrices and Markov Chains*, 2nd ed. (Springer, New-York).

[7] Z. Naor and H. Levy, Minimizing the wireless cost of tracking mobile users: An adaptive threshold scheme, in: *INFOCOM '98* (1998) pp. 720–727.

[8] C. Rose and R. Yates, Minimizing the average cost of paging under delay constraints, Wireless Networks 1(2) (1995) 211–219.

[9] C. Rose, Minimizing the average cost of paging and registration: A timer-based method, Wireless Networks 2(2) (1996) 109–116.

[10] A. Sinclair and M. Jerrum, Approximation counting, uniform generation and rapidly mixing Markov chains, Information and Computation 82 (1989) 93–133.

[11] S. Tabbane, An alternative strategy for location tracking, IEEE Journal on Selected Areas in Communications 13(5) (1995) 880–892.

Hanoch Levy received the B.A. degree in computer science with distinctions from the Technion, Israel Institute of Technology in 1980 and the M.Sc. and Ph.D. degrees in computer science from the University of California at Los Angeles, in 1982 and 1984, respectively. From 1984 to 1987 he was a member of technical staff in the Department of Teletraffic Theory at AT&T Bell Laboratories. In 1987 he joined the Department of Computer Science, Tel Aviv University, Tel Aviv, Israel. During the years 1992–1996 he has been at the School of Business and RUTCOR, Rtugers University, New-Brunswick, NJ, on leave from Tel Aviv University. His interests include computer communications networks, performance evaluation of computer systems and queueing theory.
E-mail: hanoch@math.tau.ac.il

Zohar Naor received the B.Sc. and M.Sc. degrees in computer science from Tel Aviv University, Israel, in 1987 and 1992, respectively. He has worked in the areas of digital communication and wireless networks. Currently he is working towards the Ph.D. degree in computer science at Tel Aviv University, Israel. His current research interests include location management in personal communication service networks, computer communication networks, and multiple access.
E-mail: zohar@math.tau.ac.il