Establishing a Mobile Conference Call Under Delay and Bandwidth Constraints

Amotz Bar-Noy
Computer and Information Science Department
Brooklyn College, CUNY, New York
Email: amotz@sci.brooklyn.cuny.edu

Zohar Naor
Department of Mathematics, Physics, and Computer Science
Haifa University, Israel
Email: zohar@math.haifa.ac.il

Abstract— **The issue of tracking a group of users is discussed in this study. Given the condition that the search is over only after all the users in the group are found, this problem is called the Conference Call Search (CCS) problem. The goal is to design efficient CCS strategies under delay and bandwidth constraints. While the problem of tracking a single user has been addressed by many studies, to the best of our knowledge, this study is one of the first attempts to reduce the search cost for multiple users. Moreover, as oppose to the single user tracking, for which one can always reduce the expected search delay by increasing the expected search cost, for a multiple users search the dependency between the delay and the search cost is more complicated, as demonstrated in this study. We identify the key factors affecting the search efficiency, and the dependency between them and the search delay. Our analysis shows that under tight bandwidth constraints, the CCS problem is NP-hard. We therefore propose a search method that is not optimal, but has a low computational complexity. In addition, the proposed strategy yields a low search delay as well as a low search cost. The performance of the proposed search strategy is superior to the implementation of an optimal single user search on a group of users.**

Keywords: Location management, Mobile Computing, Wireless Networks.

I. Introduction

The growing number of mobile users increases the need to support mobile groups, such as a truck fleet, a taxi fleet, and, in particular mobile Virtual Private Networks (VPN). As the number of mobile Virtual Private Networks keeps growing, the demand for tracking a group of users increases very rapidly. This is the issue addressed in this study. Given a group of users, the goal is to find all the users in the group. We call this problem the *conference call search (CCS)* problem, since the search is over only after *all* the mobile users that participate in the conference call are found.

Tracking mobile users is a key task in wireless networks. Contrary to wired networks, in which the users' locations are fixed, in wireless networks a user can potentially be located anywhere within the system service area. Whenever there is a need to route an incoming call to a particular user, the network must find the exact location of this user in order to set up the call. In this study, our concern is to efficiently utilize the wireless links associated with the tracking process. The importance of this optimization objective comes from the fact that the signaling traffic associated with tracking is transmitted through the control channel. Due to the limited bandwidth available for wireless communication, the control channel is probably the most valuable resource in a wireless network.

A. Background and related work:

The problem of reducing the cost of using the wireless communication lines while tracking mobile users has been addressed by many studies (e.g., [1], [2], [3], [5], [7], [9], [10]). Existing cellular networks use the geographic-based (GB) strategy that partitions the geographic area into zones. These zones are called *location areas* (LA) in GSM systems and *registration areas* by the IS-41 standard ([13]). The partitioning into zones is static, based on the commercial licenses granted to the operating companies. Each zone consists of a number of cells. The users register themselves in a zone whenever they enter a new zone. While moving within the zone users never register. Thus, when there is an incoming call for a specific user, all the cells belonging to its current zone need to be paged. Since a typical zone may contain a very large number of cells, the tracking cost associated with the GB strategy may be very high. Many systems incorporate the IS-41 standard with a fixed timer as follows: The user registers periodically every T time units, where T is a fixed, pre-defined parameter.

To overcome the drawbacks of the GB strategy, many methods have been proposed, aiming to reduce the wireless cost of tracking mobile users ([1], [2], [3], [5], [7], [9], [10]). The basic idea shared by these studies is that upon location change, users may or may not update their location. The criterion for a user registration may be a function of time [10], distance from last known location ([3], [9]), or number of movements between cells ([1], [3], [7]).

The studies mentioned above focus on tracking a single user. The underlying assumption for designing good searching strategies was an a-priori knowledge of the likelihood of users to be located in different cells. This knowledge is usually represented as a vector of probabilities the sum of which is 1. Each entry in the vector is associated with a user and a cell indicating the independent probability of finding this user in this cell. Optimal solutions for searching for a single user using these vectors are known for any given deadline or expected paging delay [8], [11]. For example, when the search delay is not crucial, then the optimal solution is to page the cells in a non-increasing order of their probabilities. Using a dynamic programming technique, one can find the optimal search strategy when the search must be completed after any D paging cycles, where D is a pre-defined constant.

A first step toward handling many paging requests directed to many users has been done in [12]. However, this study still concentrated on searching for a single user, under the condition that the rate of search requests is very high, such that at any given moment there is a need to find many users but the search for any user is independent of the search for other users.

The paper [4] considered the conference call search studying the trade-off between the number of paging cycles and the number of paged cells. However, bandwidth constraints were ignored by assuming that the number of paging channels available at each cell is unbounded. Moreover, the proposed search strategy is fixed in advance and cannot be changed once some of the users are found.

B. Contribution

This study addresses the CCS problem under both bandwidth and delay constraints. The objective function is to find *all* the users that belong to the group. An efficient search must take into consideration the trade-off between two basic parameters: paging cost, in terms of number of locations being searched, and paging delay. To illustrate this trade-off, let us consider the two extreme strategies for a single user search: A *blanket polling* at all the cells in the system would minimize the search delay to one paging cycle, at the expense of maximizing the expected search cost to the number of cells in the system, denoted by N. On the other hand, it has been shown in [11], that the minimum expected search cost is achieved by a sequential search in a non-increasing order of location probabilities. As a result, the expected search delay increases, while the worst case search delay is N paging cycles.

As opposed to a single user search, for which one can always minimize the search delay by increasing the search cost, the search for multiple users must take into consideration also the bandwidth constraint. That is, the number of users that the system can simultaneously search for within a cell is bounded from above by the number of available paging channels in this cell. Under tight bandwidth and delay constraints, the search must be sufficiently efficient in order to be over before the search delay deadline. Consequently, reducing the search cost does not *necessarily* increase the search delay. Under certain conditions, reducing the search cost can even reduce the expected search delay. For this reason, the objective function for efficient multiple users search is to satisfy both delay and bandwidth constraints: The search must be over during at most D paging cycles, given that at most B paging channels are available at each cell, for each paging cycle. Both D and B are pre-defined constants.

We show in this study that there is a clear trade-off between the minimal values of B and D under which a search for a group of M users is still feasible. We derive the relation among the three parameters B, D, and M. Under sufficiently large B or D, we show an optimal search strategy that minimizes the expected search cost, and its computational cost is low. We show that under tight bandwidth constraint, minimizing the expected search delay is NP-hard even for two users. We therefore propose a heuristic algorithm, that yields both a low expected search delay and a low expected search cost. Under tight bandwidth and delay constraints - we show that any multiple users search must be sufficiently efficient - in order to satisfy the delay constraint.

C. Paper organization:

Definitions and the problem formulation are given in Section 2. The CCS problem is discussed in Section 3. An efficient search strategy under general conditions is introduced in Section 4. The proposed search strategy is analyzed in Section 5. Performance analysis and numerical results are provided in Section 6. Finally, summary and concluding results are given in Section 7.

II. Definitions and Problem Formulation

Consider a cellular system consisting of a set of N cells $\mathcal{C} = \{C_1, \ldots, C_N\}$ and a set of u roaming mobile users $\mathcal{U} = \{1, 2, \ldots, u\}$. For each user $1 \leq i \leq u$, at any particular time $t \geq 0$, there exists a profile vector of length N: $V_i^t = \langle p_{i,1}^t, \ldots, p_{i,N}^t \rangle$. The meaning of this vector is that with probability $p_{i,j}^t$ user i resides at cell C_j at time t. We usually omit the superscript t and call the profile vector the *steady state probability vector*.

The *search* objective is to find a group of M users ($M \leq u$), using minimum network resources, while still maintaining acceptable average delays between search initialization and search completion. Hence, there are two constraints on the search operation: a search delay constraint, and minimizing the signaling bandwidth associated with the search operation. The search delay constraint can be reflected either by a cost associated to each search cycle delay, or by a deadline constraint - which determines the maximum search cycles. The signaling bandwidth associated with the search operations is determined by the number of the search operations, where the term "search operation" means searching for a single user within a single cell. For example, if all users locations are known to the system, then the search cost is exactly M. It is assumed that the search deadline constraint is sufficiently short, in comparison to the users' mobility, such that the search algorithm can assume that the users do not change their locations during the search process.

The general many-users search problem is to search for h users out of M for $1 \leq h \leq M \leq u$. We call this problem the *signature search (SS)* problem. We identify two special cases. When $h = M$ we call the problem the *conference call search (CCS)* problem. When $h = 1$, we call the problem the *yellow pages search (YPS)* problem. This paper studies the *conference call search (CCS)* problem.

We further distinguish between two types of search strategies. In an *oblivious strategy*, all the rounds are determined before the search process starts. The only computation allowed during the search process is to halt if all users are found. In an *adaptive strategy*, the set of paged cells in a particular round may be determined by the set of users already found in the previous rounds. At each round, the profile vector of each user is re-computed, according to the search results. In this study, we focus on the *adaptive strategy*, since this is the more efficient strategy, in terms of both search cost and search delay.

In summary, the problem we face is the following. We consider a group of M mobile users, that are roaming in a cellular network consisting of N cells. The probabilities of user i to be located in the N cells are represented by the vector V_i. The objective is to design a search strategy that minimizes the cost of finding all M users, under delay constraints (D paging cycles) and/or bandwidth constraints (B paging channels). The constraints on the time duration of the search process, may be either on the expected delay or the worst time delay. Recall that we associate one unit of cost per each search for a single user within a single cell.

III. The Conference Call Search (CCS) problem

A basic version of the CCS problem is to find all the users during at most D paging cycles, using minimum search operations. Under the condition that there is no bandwidth constraint, i.e. at each paging cycle we can search for any number of users, at any cell, the optimal search strategy is the following: For each user, apply an optimal search under deadline delay constraint of D paging cycles, using dynamic programming (see, e.g., [11]). For example, consider the case of $D = 2$ rounds. The optimal search strategy is the following: At the first paging cycle, the search is conducted at K cells. Let $P = \{p_1, p_2, \ldots, p_N\}$ be the steady state user location probability vector, sorted in non-increasing order, such that $p_1 \geq p_2 \geq \ldots \geq p_N$. p_1 is the steady state probability to find the user at the most likely location, p_2 is the probability of the second most probable location, etc. Let $P_k = \sum_{i=1}^{k} p_i$. Then, the expected search cost is given by:

$$C = P_k k + (1 - P_k)N = N - (N - k)P_k. \qquad (1)$$

Equation (1) implies that the optimal search strategy in two paging cycle is the to find the k most probable locations that minimize C. This can be found is $O(N)$ complexity for the worst case. Applying this strategy to each user is therefore an $O(MN)$ task.

We now consider the CCS problem under a bandwidth constraint, and evaluate the relation between delay and bandwidth constraints. Our goal is to find all the users, under the condition that the maximum paging channels that can be used in any cell during a paging cycle, is bounded from above by a pre-defined constant B. We show that there is a minimum upper bound on the worst case of the search delay, which is a function of B.

Theorem 3.1: Let d be the guaranteed search delay, implying that all M users can be always found during d paging cycles, but for $d - 1$ paging cycles there is no search strategy that always finds all users. Then, the following relationship holds:

$$d = \left\lceil \frac{M}{B} \right\rceil = D^*. \qquad (2)$$

Proof: Let d be the guaranteed paging delay. Using a blanket search, we can always find B users at each paging cycle, which yields a paging delay $d' = \lceil \frac{M}{B} \rceil = D^*$. Hence, $d \leq d' = D^*$. On the other hand, given users location distribution such that all users reside in the same cell, at most B users can be found at each paging cycle, due to the bandwidth constraint. Hence, under this location distribution, we get that $d \geq \lceil \frac{M}{B} \rceil = D^*$. Hence, $d = D^*$. ■

Equation 2 implies that the blanket search paging delay determines the minimum worst case deadline paging delay for any search strategy. Note that this minimum is a tight upper bound for any given M and B. However, this does not imply that the blanket search yields the minimum expected paging delay. As we show later, as the number of paging channels available for the search decreases, the expected search delay under the blanket search increases, and this strategy becomes a poor choice.

Let us consider the case where B is sufficiently large. Then, the following theorem holds:

Theorem 3.2: Given that $B \geq M$, and that all users must be found during at most D paging cycles, a search strategy that minimizes the expected search cost can be obtained with a computational complexity that is linear with M and N, for any $D \geq 1$.

Proof: Under these conditions, we can minimize the expected search cost of each user *separately*, using the methods described in [11]. Since there are M users, the number of paging channels required at any cell, for any paging cycle, is less than or equal to M. Since $B \geq M$, we can apply an optimal search for all users, without violating the bandwidth constraint, for any $D \geq 1$. It is shown in [11] that an optimal search for a single user can be obtained with a computational complexity of $O(N)$, for any given paging delay deadline D. Hence, the optimal CCS strategy under the condition $B \geq M$ has a computational complexity of $O(NM)$. ■

Theorem 3.2 implies that under the condition $B \geq M$, an optimal search can be always found using the methods described in[11]. We therefore consider the case under the condition $B < M$. Under this condition, the following theorem holds:

Theorem 3.3: Under the condition $B < M$, minimizing the expected search delay of CCS is NP hard even for two users and search delay deadline of two rounds.

Proof: For two users we get that $M = 2$. Hence, the condition $B < M$ implies that $B = 1$. The goal is to find a partition of the cells between the users in the first round to maximize some function. That is, denoting the users by p and q, we are looking for a partition $\mathcal{C} = \mathcal{C}_p \cup \mathcal{C}_q$ such that the system pages the user p in all the cells belonging to $\mathcal{C}_p$ and pages the user q in all the cells belonging to $\mathcal{C}_q$. Define, $f = \sum_{j \in \mathcal{C}_p} p_j \sum_{j \in \mathcal{C}_q} q_j$ to be the probability to find both users in the first round. Then the expected number of rounds for the strategy that uses the partition $\mathcal{C} = \mathcal{C}_p \cup \mathcal{C}_q$ is $1 \cdot f + 2 \cdot (1 - f) = 2 - f$. Therefore, the objective of an optimal strategy is to maximize f.

We now prove that finding the maximum value for f is an NP-Hard problem. The reduction is from the *subset-sum problem* which is known to be an NP-Hard problem [6]. The input to the subset-sum problem is a set of n real numbers whose sum is 1. The problem is to find a subset of the numbers whose sum is exactly $1/2$. Given the input $s_1, \ldots, s_n$ to the subset-sum problem, we consider the case where the users p and q have the same location distribution, and this location distribution is equal to the input to the subset-sum problem. Namely, the probabilities of users p and q are defined to be $p_i = q_i = s_i$ for all $1 \leq i \leq n$. Therefore, for any partition of the cells, the value of f is $x(1-x)$ for some x. The maximum of such a function is $1/4$ and is attainable if and only if $x = 1/2$. Thus, if a strategy can find the maximum value for f, then we check if this value is $1/4$. It follows that the answer is yes if and only if the answer to the original subset-sum problem is also yes. In case $f = 1/4$, we get that $\sum_{j \in \mathcal{C}_p} = 1/2$. On the other hand, we get that $f < 1/4$ if and only if the answer to the original subset-sum problem is "no". ■

Remark 3.1: Note that under the same conditions, the problem of minimizing the expected search cost using the *oblivious search strategy* is NP-hard, even for two users. In this case, the search sequence is determined before the search process starts, and the only computation allowed during the search process is to halt if all users are found. Hence, for the same example given above, the expected search cost is $N(2 - f)$, and minimizing the expected cost of the *oblivious search* can be achieved by maximizing the value of f, which is an NP-hard problem.

IV. The Sorted Round Robin (SRR) search strategy

In this section we develop a CCS strategy, under the conditions that the search for M users must be completed during at most D paging cycles, and that the maximal number of paging channels that can be used during a paging cycle within a cell is bounded from above by a pre-defined constant B. Our main goal is to reduce both the expected search cost and the expected search delay. Since Theorem 3.3 implies that under tight bandwidth condition, minimizing the expected search delay is NP-hard even for two users and two rounds, the proposed method cannot guarantee optimal performance.

The proposed strategy is based on two observations: (I) the strategy that minimizes the search cost is a search in non-increasing order of location probabilities [11], and (II) Since a CCS terminates only when the last user is found, the search delay depends only on the time duration of finding the last user. Hence, a search acceleration for some users, on the expense of neglecting the search for other users, would not necessarily reduce the expected CCS delay. Taking these observations into consideration, the following is a natural heuristic. The search attempts to conduct an optimal search for each user, i.e. to search for each user in non-increasing order of location probability. In addition, in order to minimize the search delay, we attempt to give each user its "fair share" of search bandwidth, at each round, using a round robin for user selection. Given that the remaining search delay to the search deadline is d paging cycles ($d \leq D$, where D is the search deadline constraint), in order to reduce the expected number of searches, the average number of locations that a user is paged during a single paging cycle i is given by:

$$n' = \frac{N_i}{d_i}. \quad (1 \leq d_i \leq D) \tag{3}$$

Where d_i is the number of paging cycles left until the search deadline: $d_i = D - i + 1$, where $1 \leq i \leq D$, and N_i is the number of cells yet need to be searched at the paging cycle number i. Clearly:

$$N_1 = N, \quad \forall (i > 1),\ 1 \leq N_i \leq N. \tag{4}$$

Since the maximal number of paging channels that can be used during a paging cycle within a cell is bounded from above by a pre-defined constant B, there are at most NB paging channels available at each paging cycle. Using a round robin allocation of paging channels to the M_i users that need to be found at the paging cycle number i, the number of paging channels that should be allocated to each user is given by $\frac{NB}{M_i}$. Combining this observation with Equation 3, we get that the number of paging channels that should be allocated to each user at the paging cycle number i, is given by:

$$B_u = \min \left\{ n', \frac{NB}{M_i} \right\}. \tag{5}$$

This is done by creating, for each round, a *search queue* for each user, of the B_u most probable locations to find that user, sorted in non-increasing order. Each location in which the user is paged - is deleted from the user *search queue*. Unfortunately, due to the limited bandwidth, we may face collisions between users. For example, if $B < M$, and all users have the same location distribution, an optimal search would try to search for all users at the same cells. Consequently, both targets: search efficiency for each user, and fair bandwidth allocation among all users, cannot always be achieved simultaneously. Whenever there are more than B users that need to be searched at the same location, we prefer the users with the larger probability. If the probabilities are equal - the preferred user is the one that its "share" of search bandwidth is the smaller. i.e. its *search queue* is larger. Whenever a user does not get its "fair share" of search bandwidth, due to collisions with other users, this "fair share" is complemented by the addition of the remaining most probable locations, that do not collide with other users, to the user *search queue*. As a result, the search for this user is conducted not necessarily in non-increasing order of location probabilities. Consequently, due to users location distribution overlap, we may conduct a non-optimal search for some users, while still maintaining an optimal search strategy for all other users. We denote this strategy as the Sorted Round Robin (SRR) search, since it attempts to keep an equilibrium between the users, i.e. - to conduct the same number of searches for each user, at each round. The basic idea of the SRR search is that for each paging cycle i, the system searches for each one of the M_i users need to found at the most probable B_u locations, where B_u is defined in Equation 5. For each user, the search is conducted at the most probable B_u locations in which we did not search for this user yet. In case the bandwidth at a certain location is not sufficient to search for all the M' users ($B < M' \leq M_i$) that are likely to be found in this location (e.g. a "hot spot"), the B most probable users are selected, while each one of the other $M' - B$ users is searched in the next probable location for that user, with sufficient bandwidth available for search. The SRR search is conducted as follows: For each user the system creates a queue of locations, the user *search queue*, sorted in non-increasing order of location probabilities for that user. At the paging cycle number i, $1 \leq i \leq D$, the system removes the top of each user queue, to create a queue of the users in length M_i. This queue is also sorted in non-increasing order of the users location probabilities. The search is conducted in non-increasing order of location probabilities. If there are more than B users that the system should search in a certain cell during the same round, then the B users that are the most probable to be found in this cell are chosen. The other users wait in a queue for the next round at this cell, while for the current round the search for these users would be conducted at other cells, in non-increasing order of location distribution. Users that have been found are deleted from the search list. The profile vectors of all other users (i.e. the users that are yet to be found) are re-computed, according to the results of the search. Each one from the remaining M_i users that yet need to be found gets a quota of $R' = \min\{n', \frac{NB}{M_i}\}$ locations to be searched for the next round.

Pseudo-code for the SRR adaptive search:

DATA STRUCTURES:
N cells, M users.
Search delay deadline: D.
Bandwidth constraint: B.
Search list S of users yet to be found.

INITIALIZATION:
$n' = \frac{N}{D}$.
Create the search list S, with M users.
For each user $i \in S$**:**
Create a queue Q_i of cells of length N sorted in non-increasing order of the probability to find that user in each cell.

A PAGING CYCLE:

Step 1: Compute: $R = \min\{n', \frac{NB}{M}\}$

Step 2: For each user $i \in S$, create a list L_i:
select the R most probable locations for i in which no search for i was conducted before,
and Delete these locations from Q_i.

Step 3: For each cell C_j, create a list J_j:
select the B most probable users to be searched in C_j.

Step 4: For each user i whose list L_i is not empty:
Replace each element in L_i by the most probable location C_j for i that is still in Q_i and whose list J_j is not full.
All locations in L_i that were not selected for this paging cycle are returned to the top of the queue Q_i.

Step 5: Found users are deleted from the list S.
f : number of found users in this cycle.
$M = M - f$.
IF M=0 : END OF SEARCH (Search list S is empty).
ELSE ($M > 0$):
{
$N = N - R$, $D = D - 1$, $n' = \frac{N}{D}$.
For each user $i \in S$**,** re-normalize the queue Q_i.
The summation over all elements still in Q_i equals 1.
Go to Step 1.
}

END Pseudo-code

For example, consider the case of two users p, q and four cells X, Y, Z, W. Let the probabilities be $p_X = 1/2$, $p_Y = 1/4$, $p_Z = 3/16$, $p_W = 1/16$, $q_X = 0.4$, $q_Y = 0.1$, $q_Z = 0.3$, and $q_W = 0.2$. Given that $B = 1$, in the first round the SRR search pages p in X and Y and pages q in Z and W. If p is not found in the first round, it is paged in Z and W in the second round, and if q is not found in the first round, it is paged in X and Y in the second round. The expected cost (number of paged cells) of the SRR adaptive search is therefore $4(3/4)(1/2) + 8(1/4)(1/2) + 6((3/4)(1/2) + (1/4)(1/2)) = 11/2$ cells. The search terminates after one round with probability $3/8$ when both users are found. Otherwise, with probability $5/8$ the search takes two rounds. Hence, the expected search delay is $3/8 + 2(5/8) = 13/8$ rounds. Note that due to the bandwidth constraint ($B = 1$), in the first round q is paged in Z and W even though the probability to find that user in X is larger. For this example, note that the SRR adaptive search is the optimal search.

V. ANALYSIS

The goal of an efficient CCS strategy is to reduce both the expected search delay and the expected search cost. Unfortunately, both targets may contradict each other. As we show in this section, under certain conditions the SRR search yields the minimum expected search cost, while under other conditions the SRR search yields the minimum expected search delay. Moreover, both the expected cost and the expected delay are bounded from above by the SRR search, and under tight bandwidth constraints are better than the values achieved by the blanket search strategy. This is in contrary to single user search, for which the blanket search strategy minimizes the search delay (both expected delay and worst case delay). Moreover, as the number of available paging channels decreases (B is small), the superiority of the SRR search over the blanket search increases, in terms of both the expected search delay and the expected search cost.

We now show that the SRR search does not always yield the minimum expected search cost and does not always yield the minimum expected search delay.

The following example demonstrates the first observation. Let $\varepsilon > 0$ be a very small number. Consider the following steady state profiles for all the users: $V_i = \langle 1 - (N-1)\varepsilon, \varepsilon, \ldots, \varepsilon \rangle$. Recall that$N$ is the number of cells in the system, and assume that $1-(N-1)\varepsilon \gg \varepsilon$. Given that the search delay deadline constraint is $D = \frac{NB}{M+1}$, the optimal search strategy that minimizes the search cost is to search in the first round only the locations with steady state probability $1-(N-1)\varepsilon$, and if the user is not found, to search $\frac{N-1}{D-1}$ locations at each round for the last $D-1$ rounds. On the other hand, the SRR search attempts to search for each user, at each round, at $\min\{N/D, (NB)/M\}$ locations. Indeed the expected delay of the SRR search is smaller than that of the optimal search strategy that minimizes the search cost, but its expected cost is slightly larger than the optimal cost.

The following example demonstrates the second observation. The reason for this is that at the paging cycle number r, the SRR search is conducted for each user in $\min\left\{\frac{N_r}{D_r}, \frac{N_r B}{M_r}\right\}$ locations, where N_r, D_r, and M_r are the number of cells need to be searched, the number of paging cycles until the deadline, and the number of users need to be found in round r, respectively. Given that our only goal is to minimize the expected search delay, whenever we get that $(N_r B)/M_r > N_r/D_r$, we should search for $(N_r B)/M_r$ locations. However, the SRR search prefers under this condition, to reduce the expected search cost, on the expense of a slight increase in the expected search delay, that still meets the deadline delay constraint.

The two examples shown above demonstrate that both targets: minimizing the expected search cost and minimizing the expected search delay may contradict each other. However, as opposed to single user tracking, these targets do not *always* contradict each other. While in searching for a single user, we can always trade the search cost for the expected delay, the search for a group of users is a completely different problem. Under certain conditions, we can reduce both objective targets concurrently. To illustrate this, consider a homogeneous user steady state profiles. That is, the probability of each user to be in any location is $1/N$. Assume further that $B = 1$ and that $D = M = N$. Under these conditions, the SRR search is the optimal search strategy, that minimizes both the expected cost and the expected delay. The expected cost is minimized by searching, for each user, at each round, in $N/D = 1$ location [11]. Clearly, under the given bandwidth constraint, this strategy also minimizes the expected search delay. Note that even for this example, whenever a user is found one must choose between the two targets: (I) minimum expected cost, by searching at exactly $N/D = 1$ location for each user at each round, or (II) Minimum expected delay, by increasing the number of searches for each user that has not been found yet, whenever other users are found and deleted from the search list. In this situation, the SRR search chooses to minimize the expected search cost. Given the ratio between the cost of a single search for one user, and the cost of increasing the expected delay by one paging cycle, the optimal policy that minimizes the combined cost of search cost and search delay can be found.

A. The SRR search vs. a Greedy Queue search strategy

In this subsection, we compare the performance of the SRR search strategy with the following natural greedy strategy: For each cell, create a queue of the users in a non-increasing order of their location probabilities for that cell. The initial length of each queue is M. At each round, the system pages the top B users from all queues and then delete them from the queues. In addition, all found users are deleted from all other queues. We call this strategy the *Greedy Queues (GQ)* search.

Note that the main difference between the GQ search and the SRR search, is that the GQ search does not address fairness issues. The search operations are conducted based only on the values of all the location probabilities of all users. Namely, the only parameter that counts is the success probability of the search operation. At the first sight, the GQ search looks attractive, especially when the goal function is to minimize the search cost. It even looks as a natural generalization of the result obtained in [11], that the minimum search cost is achieved by a sequential search in a non-increasing order of location probabilities. However, the GQ search is not optimal. Below we show that when the user location distribution is, for a good approximation, nearly homogeneous, the SRR search is superior to the GQ strategy.

First, We show that even for two users GQ is not optimal for both optimization objectives. Consider the case $B = 1$, $M = 2$ and $N = 4$. Denote the users p and q and denote the cells X, Y, Z, W. Let the probabilities be $p_X = p_Y = p_Z = 2/7$, $p_W = 1/7$, and $q_X = q_Y = q_Z = q_W = 1/4$. In the first round, the GQ search pages p in X, Y, Z and pages q in W. This implies that both users are found in the first round with probability $(2/7 + 2/7 + 2/7)(1/4) = (3/14)$. On the other hand, the SRR search pages p in X, Y and pages q in Z, W. This strategy finds both users in the first round with probability $(2/7 + 2/7)(1/4 + 1/4) = (2/7)$. As a result, the expected number of rounds for the GQ search is $(3/14) + 2(11/14) = (25/14) > 1.7857$, the expected number of rounds for the SRR strategy is $(2/7) + 2(5/7) = 12/7 < 1.7143$, the expected search cost for the GQ search is $4(3/14) + 7(18/28) + 5(1/28) + 8(3/28) = (179/28) > (82/14)$, and the expected search cost for the SRR search is $(4+6)(2/7) + (6+8)(3/14) = (82/14)$. In this case, the SRR strategy is the optimal search strategy, for both the adaptive search and the oblivious search, and it minimizes both the expected search delay and the expected search cost. The ratio between the expected search delay under the GQ strategy and the expected search delay under the optimal (SRR) strategy is $25/24 > 1.0416$, and the ratio between the expected search cost is $179/164 > 1.09$.

For $N > 4$, it is possible to calibrate this example for a worse ratio between the GQ strategy and the SRR strategy, in cases for which the SRR strategy is the optimal search strategy. Again, assume that $B = 1$, $M = 2$, and $D = 2$ and let $\varepsilon > 0$ be a very small number. Consider the following profiles for users p and q: $V_p = \left\langle \frac{1-\varepsilon}{N-1}, \ldots, \frac{1-\varepsilon}{N-1}, \varepsilon \right\rangle$ and $V_q = \left\langle \frac{1-2\varepsilon}{N-1}, \ldots, \frac{1-2\varepsilon}{N-1}, 2\varepsilon \right\rangle$. The GQ search pages p in the first $N-$

1 cells and pages q in the last cell C_N. The optimal (SRR) strategy pages p in the first $\lfloor (N-1)/2 \rfloor$ cells and pages q in the rest of the cells. Assume that N is odd to avoid floor operations. It follows, that with probability $(1-\varepsilon)2\varepsilon$ the GQ search finds both p and q in the first round, while the SRR strategy finds both p and q in the first round with probability $\frac{1-2\varepsilon}{2}\left(\frac{1-2\varepsilon}{2}+2\varepsilon\right)$. Hence, the expected number of rounds for the GQ search approaches 2 for a very small ε. On the other hand, the expected number of rounds for the optimal (SRR) search approaches $7/4$ for a very small ε. As a result, the ratio between the expected search delay under these strategies approaches $8/7 > 1.1428$.

B. SRR under general user location distribution

The SRR search performance depends mainly on two basic parameters: The profile vector V_i that describes the steady state location probability distribution of the user i, for each and every user, and the similarity (overlap) between the vectors V_i and V_j, associated with users i and j, for all $i \neq j$. For example, for a profile vector $V_i = (\frac{1}{N}, \ldots \frac{1}{N})$, there is no overlap to any user, since the optimal search sequence for the user i does not distinguish between different locations. On the other hand, for V_i that describes a user i that its steady state probability to reside at a specific location l is very high, the search efficiency depends on the number of other users who are more likely to be found at l, and on the steady state probabilities to find these users at l. If there are more than B users with higher steady state probability to be found at l, then the SRR search for the user i would not be optimal.

Let us define the *users overlap* $C(l,t)$ as the number of users that need to be searched at location l at paging cycle t. Note that the *users overlap* is defined per location and paging cycle. For example, under homogeneous location distribution the *users overlap* is null for all locations and paging cycles. Under a "hot spot" location distribution, when most of the users tend to reside at the same location, the *users overlap* is very high, at least for the first rounds. For example, given that the most probable location of M users is a location l_0, then $C(l_0,1) = M$. The maximum value of $C(l,t)$ for all possible values of l,t is defined as C_{max}. Clearly, for $C_{max} \leq B$, there are no collisions between the users, and the SRR search is optimal, in the sense that the search for each and every user is conducted in non-increasing order of location probabilities. Whenever $C(l,t) > B$, the paging cycle t cannot be optimal. A choice must be made between two options: A) Some users have to wait for the next paging cycle, in order to reduce the search cost on the expense of increasing the search delay, or B) The users that we could not search at l, due to bandwidth limitation, have to be paged at this paging cycle in other cells. The SRR search prefers the second option: To reduce the search delay on the expense of increasing the search cost.

The profile vector V_i affects the SRR search efficiency also in another way. It was shown in [11] that given a paging deadline delay, in order to minimize the expected search cost of a user i, the number of locations need to be searched at each paging cycle depends on the profile vector V_i. This number can be found using dynamic programming techniques [11]. The SRR search makes an approximation, by attempting to search an equal number of locations at each paging cycle. This approximation is accurate for homogeneous location distribution $V_i = (1/N, \ldots, 1/N)$, and for tight search deadline delay $D = \lceil M/B \rceil$. Hence, as the profile vector of the user "approaches" a homogeneous location distribution, the strategy of allocating an equal share of searches to each paging cycle "approaches" the optimal search strategy that minimizes the search cost. If all users have nearly homogeneous location distribution, this strategy also minimizes the search delay.

Given that the *users overlap* is negligible, that is: $C_{max} \leq B$, then the strategy of allocating an equal number of paging channels to each user is accurate either for sufficiently long search deadline delay, or for tight bandwidth constraint (e.g. $B = 1$, and $M = N$), under which the SRR search is in fact a sequential search for each user. Even if the conditions mentioned above are violated, the SRR search still makes the best effort to reduce both the expected search cost and the expected search delay. This is done by keeping a search in a non-increasing order of locations probabilities (whenever it is possible), using the minimum paging channels that still guarantee to meet the search deadline, and by allocating an equal share of paging channels to each user, at each paging cycle, in order to reduce the expected search delay.

VI. NUMERICAL RESULTS

In this Section we consider several numerical examples, that depict the performance of the SRR strategy as a function of the search bandwidth, search delay, and users location distribution. The SRR search performance is compared to the performance of two natural candidates for searching many users: A blanket search strategy, and an optimal single user search, applied on a group of users. The blanket search strategy simply searches for B users at each round, were B is the maximum number of paging channels available at each cell, for each paging cycle. The optimal single user search is applied for all users, and whenever the number of paging channels available is not sufficient, some users have to wait for the next paging cycle. We consider a system of 10 cells, and a group of 10 users. Since the SRR search is optimal for nearly homogeneous user location distribution, we consider users with non-homogeneous location distribution, such that each user is most likely to be found in a certain region within the system. We consider two cases: small users location overlap: $C_{max} < B$, and maximal users overlap: $C_{max} = M$, implying that all users are most likely to be found within one particular cell ("hot spot").

Figure 1 depicts the expected number of searches per user, as a function of the maximum (deadline) search delay, assuming non-homogeneous location distribution, and that the users location overlap is small (i.e. for each paging cycle $C_{max} \leq B$), such that there is no conflict between different users. The bandwidth constraint is taken to be the minimum B under which the search deadline D is achievable. Using

Equation 2, this value is given by: $B = \lceil M/D \rceil$. The best results are achieved by the optimal single user search strategy, that its goal is to minimize the expected number of searches, as expected. However, the search cost of the SRR strategy is, for a good approximation, sufficiently close to the performance of the optimal single user search strategy, especially for long search delay deadline. The search cost of the Blanket polling strategy is the worst, as expected. Note that since each user is most likely to reside in a certain cell or in its neighboring cells, the expected search cost per user under the SRR strategy is significantly less than the number of cells in the system. Moreover, the expected search cost decreases with the search delay deadline, as expected. Since $C_{max} \leq B$, there is no conflict between the searches for different users, and the expected search cost under the SRR strategy is very close to that achieved by the optimal single user search strategy.

Figure 2 depicts the expected search delay, as a function of the maximum search delay deadline, under the same conditions of Figure 1. The performance of the SRR search and the optimal single user search strategy are, for a good approximation, the same. The blanket search strategy has the worst expected search delay. As in Figure 1, since $C_{max} \leq B$, and each user is most likely to reside in a certain cell or in its neighboring cells, the expected search delay of the SRR strategy is significantly less than the number of cells in the system, and very close to the expected delay under the optimal single user search strategy.

Figure 3 depicts the expected number of searches, as a function of the available search bandwidth B at each cell, under the condition that all users are most likely to be found at the same cell (a "hot spot"). Hence, $C_{max} = M$ for the first paging cycle. The search deadline D is taken to be $D = \lceil M/B \rceil$. The expected search cost of the SRR strategy is much better than the blanket search strategy, but worse than the performance of the optimal single user search. The reason for this is that under large users location overlap, that satisfies the condition $C_{max} > B$, the SRR search cost is not optimal. However, the optimal single user search strategy yields a low search cost under conditions of significant users location overlap, on the expense of increasing the expected search delay. This behavior is depicted in Figure 4.

Figure 4 depicts the expected search delay, as a function of the available search bandwidth, for maximal users location overlap, under the same conditions of Figure 3. It is clearly shown that the SRR search strategy yields the best results, in terms of the expected search delay. Note that due to the users location overlap, the expected search delay under the optimal single user search is worse than that of the blanket search strategy.

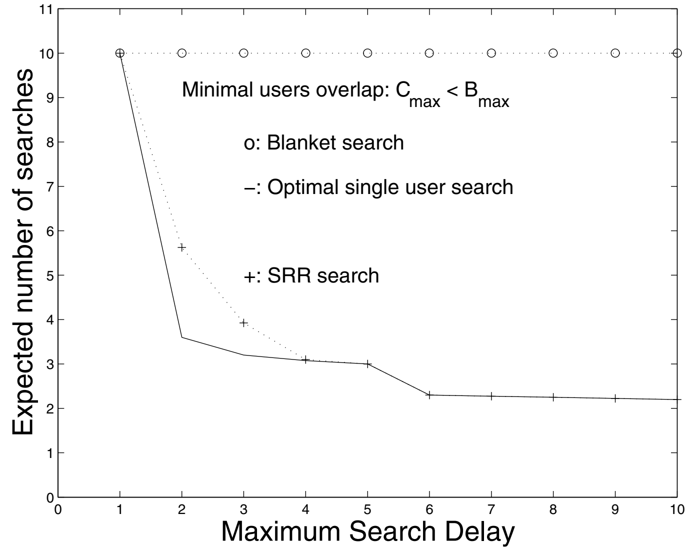

Fig. 1. The expected number of searches per user, as a function of the maximum search delay deadline, for small users location overlap.

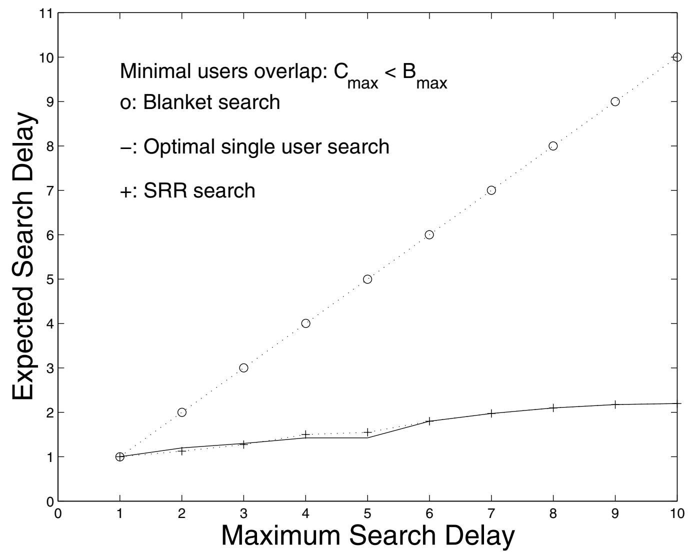

Fig. 2. The expected search delay, as a function of the maximum search delay deadline, for small users location overlap.

VII. SUMMARY AND CONCLUDING REMARKS

The issue of a multiple users search under bandwidth and delay constraints was addressed in this study. We showed that under tight bandwidth and delay constraints, the problem of finding an optimal search strategy is NP-hard. In addition, we showed that as opposed to single user tracking, for which we can always reduce the expected search delay by increasing the search cost, the dependency of the expected cost of searching *many* users on the expected delay is more complicated, and depends on the bandwidth constraint. We therefore proposed a search strategy, called SRR search, aiming to reduce the combination of both the search cost and the search delay. The SRR search is optimal under conditions of nearly homogeneous users location distribution. Nevertheless, our analysis and numerical results show that it performs very well under general conditions of tight bandwidth and delay constraints. Numerical and analytical results show that the SRR search performs significantly better than the blanket search strategy, and its expected delay is better than the expected delay of a single user optimal search, applied to a group of users. We

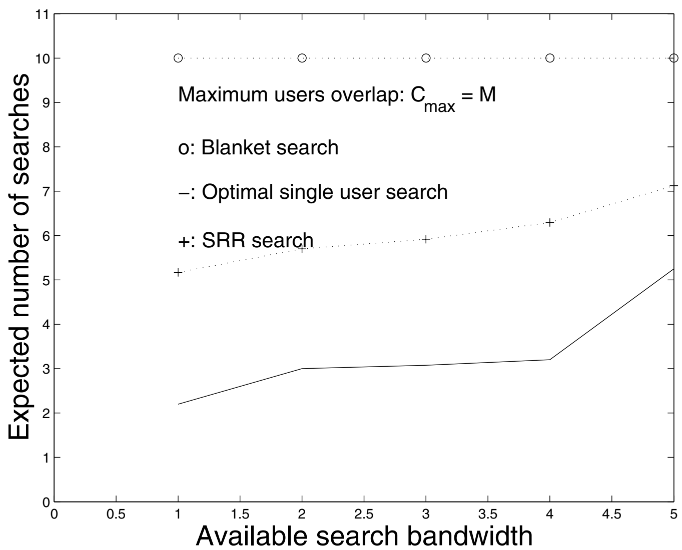

Fig. 3. The expected number of searches, as a function of the available search bandwidth, for maximal users location overlap.

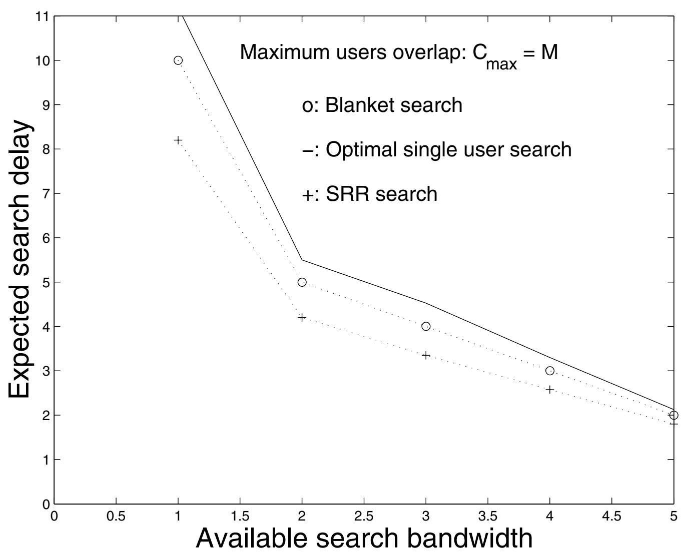

Fig. 4. The expected search delay, as a function of the available search bandwidth, for maximal users location overlap.

also showed that the SRR search outperforms the natural GQ search strategy.

REFERENCES

[1] I. F. Akyildiz, J. S. M. Ho, and Y.B. Lin. *Movement Based Location Update and Selective Paging Schemes.* IEEE/ACM Trans. On Networking, 4(4):629–638, 1996.

[2] I. F. Akyildiz, J. McNair, J. S. M. Ho, H. Uzunalioglu, and W. Wang. *Mobility Management in Next Generation Wireless Systems.* IEEE Proceedings Journal, 87(8):1347–1385 ,1999.

[3] A. Bar-Noy, I. Kessler, and M .Sidi. *Mobile Users: To Update Or Not to Update?* Wireless Networks, 1(2):175–185, 1995.

[4] A. Bar-Noy and G. Malewicz. *Establishing Wireless Conference Calls Under Delay Constraints.* The 21th ACM Symp. on Principles of Distributed Computing (PODC), pp. 41–50, 2002.

[5] A. Bhattacharya and S.K. Das. *LeZi-Update: An Information-Theoretic Framework for Personal Mobility Tracking in PCS Networks.* Wireless Networks 8(2-3):121-135, 2002

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman, 1979.

[7] J. S. M. Ho and I. F. Akyildiz. *Mobile User Location Update and Paging Under Delay Constraints.* Wireless Networks, 1(4):413–425, 1995.

[8] S. Madhavapeddy, K. Basu, and A. Roberts. *Adaptive Paging Algorithms for Cellular Systems.* Wireless Information Networks: Architecture, Resource Management and Mobile Data, 83–101 ,1996.

[9] U. Madhow, L. Honig, and K. Steiglitz. *Optimization of Wireless Resources for Personal Communications Mobility Tracking.* IEEE Trans. on networking, 3(6):698–707, 1995.

[10] C. Rose. *Minimizing the Average Cost of Paging and Registration: A Timer-Based Method.* Wireless Networks, 2(2):109–116, 1996.

[11] C. Rose and R. Yates. *Minimizing the Average Cost for Paging Under Delay Constraints.* Wireless Networks, 1(2):211–219, 1995.

[12] C. Rose and R. Yates. *Ensemble Polling Strategies for Increased Paging Capacity in Mobile Communication Networks.* Wireless Networks, 3(2):159–167, 1997.

[13] EIA/TIA. *Cellular Radio-Telecommunications Inter-system Operations.* Tech. Rep. IS-41 Revision C, 1995.