Optimal Paging for Mobile Location Tracking [1]

Farooq M. Anjum [2], Leandros Tassiulas[3] and Mark Shayman [4]
Dept. of Electrical Engineering
University of Maryland
College Park, MD 20742
fanjum@telcordia.com (leandros, shayman)@isr.umd.edu,

Abstract: For applications such as personal wireless communications which support mobility of users , locating the mobiles efficiently is critical. In this paper we look at the problem of paging in such a multiuser environment where information on the mobility pattern of the user is assumed available. We characterize the region of paging request generation rate vectors for which there exists a paging policy such that the system does not become unstable and give an existential condition for the stable paging request generation rate vectors. A policy which achieves the maximum paging requests processing rate is also specified. Another important result shown is that in order to attain the maximum discovery rate in a delay insensitive system, it is sufficient to concentrate on restrictive paging policies whereby each mobile is paged in only one cell during a time slot. In other words, blanket paging does not improve the discovery rate of a delay insensitive paging system. Results obtained are applicable as performance limits for the operation of the paging policies in delay sensitive systems such as cellular telephony.

1 INTRODUCTION

Consider a wireless communication network with cellular structure and many mobiles. During the operation of the system when a mobile user has to be contacted either to deliver a call to it or to deliver data, the base station which can contact the mobile user is determined and thereby the call is completed by

[1]A partial version of this paper appeared in the International Teletraffic Congress June97 Washington DC.

[2]Currently with Telcordia Tech. Inc, Morristown NJ

[3]This research was supported in part by NSF under grants NCR-9211417, NCR-9406415, ECS-9626399 and AFOSR under grant 95-1-006.

[4]This research was supported in part by NSF under grant ECS9626399 and DoD contract number MDA90497C3015

setting up a connection between the mobile user and the chosen base station. Since a mobile does not have a fixed location, a problem that arises in order to ensure connectivity of mobile users is that of keeping track of the location of the users.

One option that arises at this point is for the system to keep track of every move that the user makes so that the problem of determining the nearest base station is easily determined. This may be done by requiring the user to register with the system every time it moves out of the coverage area of a base station. But, as is obvious, this may lead to unneccessary waste of system resources and may even lead to overloading of the system especially if the system has mobiles which receive calls infrequently and move about frequently. The other alternative would be to let the mobiles roam throughout the system and whenever the mobile user is to be contacted, for the system to page it in all the cells till the user is found. This option while reducing registration costs, would increase tremendously the paging costs since each mobile may typically be have to be paged in many cells before it is found.

Thus, the problem of location management consists of determining schemes such that the overall cost of locating mobile users when necessary is reduced. A location management scheme consists of two parts namely registration and paging. Registration is the process whereby the mobile informs the system of its location and paging is the process whereby the system tries to infer the exact location of a mobile on the basis of the information available through the registration process and other statistics.

The registration strategy currently in use in cellular telephony is based on the concept of *registration areas*. The collection of all base stations (cells) in the system is partitioned into a number of disjoint groups called as registration areas. Each cell broadcasts the ID of the registration area to which it belongs. Whenever a mobile crosses the boundary of two cells belonging to different registration areas, it registers itself to the cell that is entered. Hence at every time instance the system is aware of the registration area to which the mobile belongs, and therefore the mobile need not be paged in any cell outside the most recent registration area. Now if the mobile registers whenever it changes a cell, then the system would know the exact mobile location. In such a case, paging would be unneccessary leading to no paging costs and high registration costs. At the other extreme if the mobile user never registers, then the registration costs would be minimum while the paging costs would be high. An optimal solution which minimizes the total cost lies between these two extremes [1, 2, 4, 7, 8]. Hence, the registration areas in current PCS networks consist of many base stations or cells.

The goal of the paging process is to obtain the precise location of the user.

Most of the current PCS systems resort to blanket polling, in which all the cells present within the location area of the mobile are polled simultaneously for a single mobile. Other schemes have also been suggested [5] such as a two step procedure whereby initially a group of preferred locations are polled for a mobile. If this is unsuccessful in locating the mobile, then blanket polling is to be resorted to. But schemes such as these are clearly inefficient since the amount of resources spent in locating a single mobile is clearly very high.

A paging strategy determines in which cells a mobile will be paged and in what sequence, until it is discovered. A good paging strategy identifies the location of the mobile while it uses the paging channels efficiently. The paging controller determines the paging strategy for each mobile based on the likelihood that the mobile is located in each cell. This is determined in turn based on location information about the mobile, obtained during *registration*, as well as the statistics of the mobility pattern of the mobile, if known. The paging controller is faced with the following problem. Requests for locating mobiles are generated according to some arrival process and they are buffered until the mobile is located. Based on the registration information and the mobility pattern, the location probability distribution of the mobile is generated. Based on the location probability distribution of the mobiles currently in the buffer, the paging controller determines which mobiles will be paged in each cell. If the number of pending requests is smaller than the number of available channels in each cell, then each mobile can be paged simultaneously in all cells and it will be located in one slot. If the number of pending requests is large, each mobile may be paged in only a few cells at a time. In this manner the discovery rate can be increased significantly as we will see later.

Thus, we consider a wireless network with N cells and a large number of mobile users. At each time instant a mobile is residing in one of the N cells. If it is paged by the base of that cell, it will respond to the paging by a registration message and its current location will be identified by the system. Since the mobile is moving, the system in general does not know the cell where it resides at each time instant. During the operation of the system, the location of a mobile user needs to be identified at certain times such as when a call has to be delivered to the user. At these times the mobile is paged in different cells until it is paged in the cell where it resides in which case its location is identified by the system.

It has been shown in [7] that there is a lot of improvement in the number of paging requests discovered in a given time interval, which we call the discovery rate, with sequential paging, in which the mobiles are paged first in the location where the probability of finding them is the highest and paged last in the locations where the probability of finding them is the least, as compared to blanket paging. As a result, with sequential paging, the number of cells in which the mobile is paged is far less than with blanket paging. But a problem is that sequential paging does not generalize in case of a multiuser environment with arbitrary location distributions of the mobiles. An attempt to adapt paging policies for single user to a multiuser environment has been done in [8]. In [3, 6, 12] the authors present different paging strategies for a multiuser environment and attempt to analyse the performance of different paging strategies in terms of delay and the number of messages transmitted. No effort has been made earlier either to determine the maximum number of paging requests which can be sustained by a system given a certain number of resources in the form of paging channels or at investigating optimal paging policies.

The basic motivation for this work is to come up with paging policies which locate the maximum possible paging requests given certain number of paging channels and mobiles with arbitrary probability location distributions. We ignore delay for this study and hence the results can be considered to be applicable as performance limits in delay sensitive systems such as cellular telephony or can be directly used in case of delay insensitive systems such as messaging [11]. Towards this end, we first characterize the maximum achievable rates of mobile location discovery through paging for a given paging channel capacity. For rates more than the achievable rates we show that the system is unstable, *i.e* the queues tend to increase without bound for any paging policy. We next specify a strategy called the maximum throughput policy, that stabilizes the system for any of the achievable discovery rates. The issue of simultaneous pagings in more than one cell for the same mobile is also addressed. We show an important result that the maximum possible paging requests that can be sustained by a delay insensitive system does not depend on how many cells are paged simultaneously for a given user. Thus, for a delay insensitive system, the maximal achievable paging rate is the same, for both a system in which mobiles are paged in one cell at a time as well as for a system in which a mobile user is paged in more than one cell at a time.

The paper is organized as follows: In section 2 a queuing network representation of the actual wireless network is described. Doing so, we intend to structure the problem faced by the paging controller of determining which mobile to page in which cell and when, while minimizing the use of the system resources as a queuing theory problem. Using the given queuing network representation, in section 3 we characterize the set of paging request generation rate vectors for which the system can achieve satisfactory performance. A policy which stabilizes the system for any such paging request generation rate vector is also given. We also show in the same section, that it is enough to consider paging policies which are restrictive in the sense of precluding simultaneous paging of a mobile in more than one cell if the aim is to maintain the performance of the system in the sense of achieving the maximum discovery rate. The given policy, however suffers from the curse of dimensionality. In section 4 we provide a paging policy for a system consisting of two cells which, as we show through simulations, achieves the optimal location request rates.

2 THE MODEL

Each mobile registers itself whenever it is paged as well as at other time instants determined by its registration strategy. At each time instant the system which is assumed to have N cells has some statistical information about the location of each mobile, that is the distribution of the mobile location conditional on the previous registrations and based on the mobility pattern of the mobile. Location probability distribution of a mobile to be located is assumed to be one of K possible distributions denoted as p^k, $k = 1, .., K$, where $p^k(m)$ is the probability that the mobile of class k is in cell m, $m = 1, .., N$. Every time a request to find some mobile is generated, the conditional probability distribution of the mobile locations will be one of those K distributions. Those K distributions might represent for instance the registration area where the mobile resides. Thus, the location probability distribution is quite general and hence might also be derived from the recorded history of user mobility.

Depending on the registration strategy the initial location distribution may have other interpretations as well.

It is assumed that from the time a request for a mobile starts to be processed until the mobile is actually discovered, the cell where it resides does not change. Therefore a mobile needs to be paged at most once in each cell until it is discovered. This assumption is valid because the processing of a request happens much faster than the rate at which the mobile crosses cells. Therefore, the likelihood that the mobile changes cells while the request is being processed is very low.

Let us denote the collection of all cells in the system by $Z = \{1, \cdots, N\}$. For any subset of cells S, denote by p_S^k the kth location distribution restricted in the set S, that is

$$\begin{aligned} p_S^k(m) &= \frac{1}{\sum_{i \in S} p^k(i)} p^k(m) \quad \text{if} \quad m \in S \\ &= 0 \qquad \text{otherwise} \end{aligned}$$

Note that with this notation, for any cell i, $p^k(i) = p_Z^k(i)$. Also define $p_S^k(S') = \sum_{\forall i \in S'} p_S^k(i)$.

In order to make things concrete, consider for example a two layer overlapping registration system with N cells and 3 registration areas, S_1, S_2 and S_3. Each cell belongs to two registration areas due to which this is called the overlapping registration system. It has been shown in [9] that the registration costs for such a system are significantly lower than for a system with disjoint registration areas. In such an overlapping registration system, consider three sets of mobiles set 1, set 2 and set 3. At a particular time let a mobile belonging to set 1, set 2 and set 3 be in S_1, S_2 and S_3 respectively, and assume that the mobiles have to be located since a call has to be delivered to them. In the absence of other information, assume that each mobile has a uniform location probability over the cells of the location area in which it is registered. Thus the probability distribution for each mobile is then completely specified. Now when the paging controller has to decide which mobile to page in which cell, it has to take into consideration the different location probabilities for each of the mobiles. For instance, to decide which mobile, of set 1 or set 2, to page in the cells common to both S_1 and S_2, the paging controller should take into consideration the location probabilities of each mobile set. Thus by doing so the number of mobiles found in a given amount of time can be increased which implies a more efficient use of the system resources.

Let us now look at the actual physical system, the model of which is shown in figure 1. Paging requests arrive at the paging controller and are queued. Based on the paging algorithm in use, the paging controller decides which mobile to page in which cell and when. If the mobile is found, the corresponding paging request is deleted from the queue. But if the mobile is not found, it may be paged in other cells in the next time slot as decided by the paging controller. For example in case of the *blanket polling* strategy, the paging controller decides to page that mobile the request for which is at the head of the queue in all the cells. Hence, assuming a single paging channel in all the cells, in figure 1, we would have P1, P2, ... PN being all equal to the same mobile. Further, the mobile is found as a result of blanket polling and hence the paging request of the mobile is deleted from the queue. In terms of the example above, assuming that there is a single paging controller for all the cells in the system (though variations on this are easily possible) the paging controller has three requests corresponding to mobiles of set 1, set 2 and set 3 above. So in order to decide which cells to page in the cells common to two registration areas, the paging controller should base its decision also considering the location probabilities. Thus it is quite obvious that the paging controller can improve the performance achieved from blanket polling strategies by following more intelligent strategies which make use of the information known, namely the probability statistics of the mobile. Note that in this case, the paging controller may not be able to follow a sequential paging policy because of conflicts between two mobiles for the same cell, resolution of which may cause one mobile to stick to the sequential policy while the other mobile has to necessarily deviate from the sequential policy.

So, going back to the model, consider the location probability distribution of a mobile. When a mobile with distribution p^k_S is paged in some set of cells S' at slot t, it may be discovered with probability $p^k_S(S')$. If it is not discovered, then its location probability distribution at slot $t+1$ becomes $p^k_{S-S'}$. Denote by $X^k_S(t)$ the number of pending requests for mobiles with distribution p^k_S at slot t. Assume that requests for locating mobiles with location distribution p^k are generated according to a Poisson process of rate a_k. Further, there are l_m paging channels in each cell m. Hence, at each slot in cell m at most l_m mobiles can be paged. Which mobile will be paged in each slot is determined by the paging policy.

In the paging controller, there are paging requests at different processing stages depending on, in which cells the mobiles have already been paged. After each paging slot some mobiles are discovered and some other proceed to a different processing stage. The paging controller can now be modelled logically as a network consisting of X nodes and Y links. The connectivity of the system is represented by the directed graph $G = (X, Y)$ where X is the set of nodes and Y is the set of links. A node represents a subset of the cells $Z = \{1, \cdots, N\}$ and the requests residing at the node in the logical system correspond to a mobile residing at any of those cells in the actual system. Thus there are as many nodes in the network as the number of possible subsets of Z. A subset of cells corresponds to each link as well. A mobile traverses a link when it is paged in the cells that correspond to the link and is not discovered thus moving to the next stage. From each node emanate as many links as the number of possible subsets of the cells constituting that node. At each node S, the requests for mobiles to be paged are queued up such that the queue X^k_S consists of paging requests of mobiles for which $p^k_S(m) = 0 \ \forall m \notin S$ and $p^k_S(m) \neq 0 \ \forall m \in S$. Hence each link can now be looked on as a server such that it can serve or page the mobile requests queued up at its origin node depending upon constraints on capacity. Note that due to the limited number of paging channels at each cell, it is not possible to serve all links at every slot. Rather activation of certain links prevents activation of others. After being paged (served), a request either leaves the system if the mobile is found or is directed to the destination node of the link. The logical queuing of mobile requests at different nodes corresponds to the buffering of location requests in the actual paging controller. A node in the logical system corresponds to the set of cells over which the mobile has a non-zero probability of being found in the actual system. Thus, every mobile request in the actual system is queued up at a node in the logical system. Paging

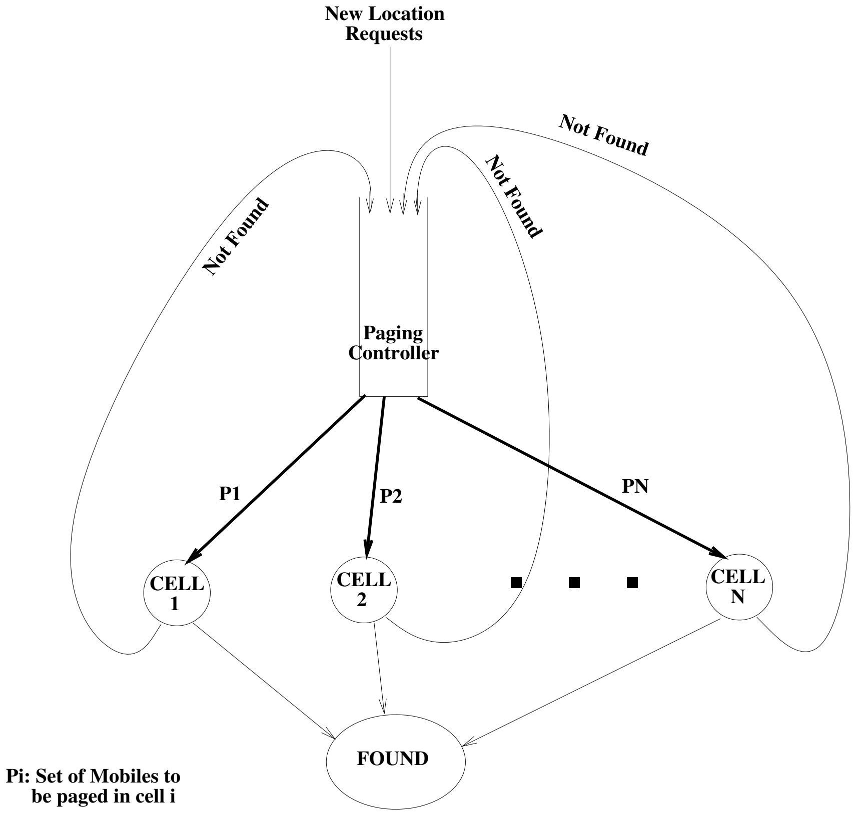

Figure 1: *The actual model.*

of a mobile in one or more cells in the actual system corresponds in the logical system to the traversing of a link representing the set of cells over which the mobile is paged with the link emanating from the node at which the request is queued. If the mobile is found then the request leaves the system, else the request is queued at the destination node in the logical system which now represents the new set of cells over which the mobile has a non-zero probability distribution. This new set of cells does not consist of those cells over which the mobile has been paged and not found.

Assuming that during a time slot requests can traverse at most one link and also assuming that each link represents paging in only one cell there would be 2^N nodes and $(2^{N-1}-1)N$ links in the system. Relaxing the latter assumption in order to make the modeling more general greatly increases the number of links.

At this point without loss of generality we also assume that arrivals from

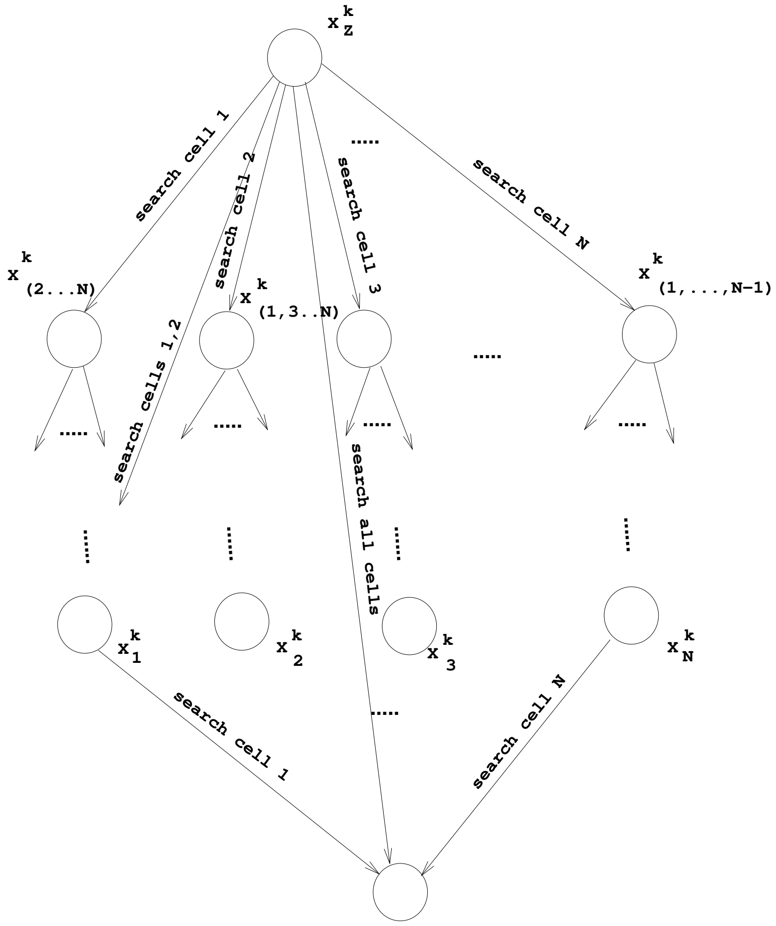

Figure 2: *The logical queuing model.*

outside the system in the form of paging requests are not allowed to queue at any node randomly on entering the system. Rather each class of mobiles has associated with it, its own entrance node S^k which corresponds to the set of nodes over which mobiles of class k have a non-zero location probability . This is consistent with the fact that mobiles of the same class have the same probability distribution p^k. Thus, the entrance node of class k, S^k, consists of cells $m, \;\; m \in Z$, such that

$$p^k(m) \neq 0 \;\forall m \in S^k \text{ and } p^k(m) = 0 \;\forall m \notin S^k$$

Thus the problem faced by the paging controller is now converted to the determination of the class of mobile and the node and link over which a mobile of this class is to be paged. The configuration of pagings performed in slot t is represented by the collection of variables $G(t) = \{g^k_{SS'}(t) : k = 1, \cdots, K, S' \subset S \subset Z\}$ where $g^k_{SS'}(t)$ is the number of mobiles of class k restricted in S that are paged in cells S' at slot t. A paging configuration $G(t)$ is feasible if

$$\sum_{k=1}^{K} \sum_{S \subset Z} \sum_{S' : m \in S' \subset S} g^k_{SS'}(t) \leq l_m \qquad m \in Z$$

where l_m is the number of paging channels in cell m.

Let us denote by $\mathcal{G}$ the collection of all feasible paging configurations. A paging policy is any sequence of feasible configurations $G = \{G(t)\}_{t=1}^{\infty}$. In the sequel we consider only policies where $G(t)$ is a function of the current backlog of pending requests in the system and are thus stationary with respect to time. Thus we now seek to identify the request generation rate vectors, $\mathbf{a} = (a_1, \cdots, a_K)$ which can be served by the paging controller in steady state ($i.e$ without request backlog becoming unstable) as well as paging strategies which achieve that.

3 CHARACTERIZATION OF ACHIEVABLE PAGING REQUEST GENERATION RATE VECTORS

In this section we proceed to characterize the maximal set $\bar{C}$ of discovery rate vectors. We also give a policy which as we show, achieves the maximum discovery rate and hence call it the maximum throughput policy. Then we show that the existence of an achievable discovery rate vector is closely linked to a condition stated below. A few lemmas precede the major results stated as theorems.

Let $f^k_{SS'}$, $S' \subset S \subset S^k$ denote the long-term average flow of requests from queue X^k_S to queue $X^k_{S-S'}$. Hence the flow of requests out of queue X^k_S along the link corresponding to S' is given by $(1/p^k_S(S-S'))f^k_{SS'}$ and the flow of requests out of queue X^k_S leaving the system since the corresponding mobile is discovered in a cell of S' is $(p^k_S(S')/p^k_S(S-S'))f^k_{SS'}$. Also the aggregate flow into queue X^k_S comprises of $\sum_{S':S' \cap S = \emptyset} f^k_{(S \cup S')S'}$ and possible exogeneous arrivals.

Definition 1 *The system is stable if the queue length process reaches steady state and does not become unbounded under the given policy.*

Definition 2 *A given paging request generation rate vector* $\mathbf{a}$ *is achievable if there exists a policy* $\tilde{G} \in \mathcal{G}$ *such that the system is stable under* $\tilde{G}$ *for* $\mathbf{a}$.

Now consider the following condition, called A in the sequel.

A. For every class $k = 1,..,K$ and set of cells S there are nonegative numbers $f^k_{SS'}$, $S' \subset S$ such that

$$\text{A1}: \quad \sum_{k=1}^{K} \sum_{S} \sum_{S':m \in S'} \frac{f^k_{SS'}}{p^k_S(S-S')} \leq l_m \tag{1}$$

and

$$\text{A2}: \quad \sum_{S' \cap S = \emptyset} f^k_{(S \cup S')S'} = \sum_{S' \subset S} \frac{f^k_{SS'}}{p^k_S(S-S')} \text{for the non-entry cells} \tag{2}$$

and we have for the entry cells S^k given the discovery rate vector as $\mathbf{a} = \{a_i : i = 1 : K\}$

$$\text{A2}: \quad a_k + \sum_{S' \cap S = \emptyset} f^k_{(S \cup S')S'} = \sum_{S' \subset S} \frac{f^k_{SS'}}{p^k_S(S-S')} \tag{3}$$

A1 represents the constraint imposed due to the limitation on paging channels in cell m, while A2 represents flow conservation in each queue X^k_S under steady state conditions.

Lemma 1 *If a discovery rate vector* $\mathbf{a} = (a_i : i = 1,..,K)$, *is achievable in* $\mathcal{G}$,*then there exist flow rates such that condition A holds.*

Proof:

Let $f^k_{SS'}(t)$ be the number of mobiles in X^k_S which move to $X^k_{S-S'}$ and let $e^k_{SS'}(t)$ be the number of mobiles in X^k_S which are discovered after they are paged in $S^{'}$ during slot t. Note that

$$f^k_{SS'}(t) + e^k_{SS'}(t) \quad = \quad g^k_{SS'}(t)$$

Let us define

$$\begin{aligned} f^k_{SS'} &= lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} f^k_{SS'}(\tau) \\ e^k_{SS'} &= lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} e^k_{SS'}(\tau) \end{aligned}$$

Since $\mathbf{a}$ is achievable and the paging channels are limited, the limits exist. Note that we assume that it is not possible for $f^k_{SS'}(\tau)$ and $e^k_{SS'}(\tau)$ above to fluctuate which is justified for any practical paging policy. Further, using the fact that we have an ergodic process,

$$f^k_{SS'}/e^k_{SS'} = \frac{p^k_S(S-S')}{p^k_S(S')}$$

Since the paging configuration $G(t)$ is feasible, it follows that

$$\sum_{k=1}^{K} \sum_{S\subset\{1,\cdots,N\}} \sum_{S':m\in S'} f^k_{SS'}(t) + \sum_{k=1}^{K} \sum_{S\subset\{1,\cdots,N\}} \sum_{S':m\in S'} e^k_{SS'}(t) \;\; \leq \;\; l_m$$

which implies that

$$\sum_{k=1}^{K} \sum_{S\subset\{1,\cdots,N\}} \sum_{S':m\in S'} \frac{f^k_{SS'}}{p^k_S(S-S')} \;\; \leq \;\; l_m$$

which implies that constraint A1 is satisfied. Furthermore at steady state, at the entry node $S = S^k$ of class k we have

$$\sum_{\tau=1}^{t} a_k(\tau) + \sum_{\tau=1}^{t} \sum_{S':S\cap S'=\emptyset} f^k_{(S\cup S')S'}(\tau) = X^k_S(t) + \sum_{\tau=1}^{t}\sum_{S'} f^k_{SS'}(\tau) + \sum_{\tau=1}^{t}\sum_{S'} e^k_{SS'}(\tau)$$

and at the nonentry nodes $S \neq S^k$ we have,

$$\sum_{\tau=1}^{t} \sum_{S':S\cap S'=\emptyset} f^k_{(S\cup S')S'}(\tau) = X^k_S(t) + \sum_{\tau=1}^{t}\sum_{S'} f^k_{SS'}(\tau) + \sum_{\tau=1}^{t}\sum_{S'} e^k_{SS'}(\tau)$$

Dividing by t and taking limits we see that A2 holds.

$\diamond$

3.1 Maximum throughput policy

Policy π that we specify next achieves maximum discovery rate in the wireless network being considered. The logic behind the policy is to give weights to every class of mobiles over every link in the logical queuing network and thereby decide the best class of mobiles for each link. The weight of this class is now considered the weight of the link. Using the given link weights, we next select links over which mobiles are to be paged while satisfying the capacity constraint. This is an adaptation of the policy given in [10] for our problem. However, our problem differs from that considered in [10] in two ways. Firstly, in our case the set of destination nodes for any class is not fixed but is probabilistic.

Secondly, the constraint vector c, defined in stage 2 below, is not a binary vector but a nonnegative integer-valued vector. It is to be pointed out that π is a stationary function of the queue lengths. Noting these differences, we determine the stationary activation rule for π denoted by $g(.)$ in three stages as given below.

Stage 1: For each server i, a weight $D_i(t)$ is selected as follows. For each class k define $D_i^k(t)$ by

$$D_i^k(t) = X_S^k(t-1) - X_{(S-S')}^k(t-1).p_S^k(S-S') \tag{4}$$

where link i connects node S to node $S-S'$ and hence represents paging over the set of cells S'.

Let $D_i(t) = max_{k=1,\cdots,K}\{D_i^k(t)\}$ be the weight of server i and $D(t) = \{D_i(t) : i = 1, 2, \cdots\}$ be the weight vector at slot t.

Thus in this stage, we try to find the best mobile class to page, at every possible combination of cells in the system, if that combination of cells were selected for paging by the paging controller. Each class is examined in terms of the weight $D_i^k(t)$ above.

Stage 2: Let c be a vector with as many elements as the number of links. c represents the individual capacity of each link in the set of links which can be activated in the same slot. The constraint on c being that, the summation of the capacities of all links representing paging in sets that include the cell m is bounded by l_m, the number of paging channels in m. Define

$$\hat{c} = arg\ max_{\forall c}\{D^T(t)c\}$$

In case of a choice pick any vector c which maximizes the above expression.

In this stage the paging controller tries to pick the set of cells over which to page given the constraint on paging channels. This is based on the weight given to each link in the previous stage.

Stage 3: Let $\hat{k}_i$ be the class for which $D_i(t) = D_i^k(t)$ for each server i. In case of a choice pick one. Now let

$$\begin{aligned} g_{SS'}^k(t) &= \hat{c}_i, \text{ if } k = \hat{k}_i \text{ and } X_S^k(t-1) \\ &\quad \text{exceeds the capacity of the links to be activated at node S in slot t} \\ &= 0 \text{ otherwise} \end{aligned}$$

Finally, in this step the paging controller decides how many mobiles to page over the links chosen. The mobiles being paged belong to the class chosen in stage 1 for each of the links chosen in stage 2. Note that as a result of these three stages, the sets $Pi \ \imath = 1, \cdots, N$ in figure 1 are decided and the paging is then carried out.

Equation (4) can be viewed as one obtained by taking an expectation of the corresponding equation in [10] and it is necessitated by the probabilistic nature of the destination nodes. In fact, it also represents the one step evolution of the system provided a request of class k is paged on the set of cells S'.

We have seen that the existence of a discovery rate vector ensures that condition A is satisfied. Now we characterize the set which consists of all the discovery rate vectors for the given system.

3.2 Characterization of achievable discovery rate vectors

The vector of nonnegative numbers, $f^k = \{f^k_{SS'} : \forall S' \subset S \subset Z\}$ is called an admissible flow vector for class k if it satisfies constraint A2. Now $f = \{f^k : k = 1, \cdots, K\}$ a matrix consisting of non-negative numbers such that the corresponding f^k satisfy constraint A2 and f satisfies constraint A1 as well is called an admissible multicommodity flow vector. Let F_a be the set of all admissible multicommodity flow vectors associated with the flow vector $\mathbf{a}$.

Let

$$\mathbf{C}'_{\mathbf{a}} = \{\mathbf{a} : \exists f \in F_a, \text{such that A1 is satisfied as a strict inequality}\}$$

Further the closure $\bar{\mathbf{C}}'_{\mathbf{a}}$ of $\mathbf{C}'_{\mathbf{a}}$ is as follows [10]

$$\bar{\mathbf{C}}'_{\mathbf{a}} = \{\mathbf{a} : \exists f \in F_a\}$$

Now we state a lemma based on the sets $\mathbf{C}'_{\mathbf{a}}$ and $\bar{\mathbf{C}}'_{\mathbf{a}}$ constructed above.

Lemma 2 *The given policy π is such that the system is stable for every $\mathbf{a} \in \mathbf{C}'_{\mathbf{a}}$. Further, if $\mathbf{a} \in (\bar{\mathbf{C}}'_{\mathbf{a}})^{\mathbf{c}}$, then the system is unstable for any policy, both in the class of stationary and nonstationary policies with respect to the queue lengths.*

Proof: In order to prove the lemma we would need to look at the system dynamics. Let us define a random routing matrix R^k for class k with r^k_{ij} as its elements, where i refers to a node and j refers to a link. Also let $o(j)$ and $d(j)$ refer to the origin and destination nodes of link j. With this let,

$$\begin{aligned} r^k_{ij} &= -1 \text{ if } o(j) = i \\ &= 1 \text{ with probability } p^k_{o(j)}(d(j)) \text{ if } d(j) = i \\ &= 0 \text{ otherwise} \end{aligned}$$

As a result, the equation governing the evolution of the system state can be written as

$$X^k(t+1) = X^k(t) + R^k g^k(t) + A^k(t+1) \tag{5}$$

where $X^k(t) = \{X_S^k(t), \forall S\}$, $g^k(t) = \{g_{SS'}^k(t), \forall S' \subset S\}$ and $A^k(t) = \{A_S^k(t)\ \forall S\}$ $A_S^k(t) = a_k$ if $S = S^k$, otherwise equals 0.

Using (5) and the maximum throughput policy as defined, the rest of the proof follows from [10]. Hence we omit it for the sake of brevity. As a result, it can be concluded that the policy π is a maximum throughput policy.

$\diamond$

Based on this lemma we have the following theorem.

Theorem 1 *Let $\bar{C}$ be the maximal set containing all the paging request generation rate vectors using the unrestrictive multi-stage policies. Then we have*

$$\mathbf{C}'_{\mathbf{a}} \subset \bar{C} \subset \bar{\mathbf{C}}'_{\mathbf{a}}$$

Note that in case $\mathbf{a}$ satisfies A1 and A2, then $\mathbf{a} \in \bar{\mathbf{C}}'_{\mathbf{a}}$. But, it can be easily argued that $\bar{\mathbf{C}}'_{\mathbf{a}} - \mathbf{C}'_{\mathbf{a}}$ is the boundary of $\mathbf{C}'_{\mathbf{a}}$ which is a surface and has no interior in the space of paging request generation rate vectors $\mathbf{a}$. Hence Theorem 1 characterizes $\bar{C}$ up to a set of Lebesgue measure zero. In fact, for all practical purposes $\mathbf{a}' \in \bar{\mathbf{C}}'_{\mathbf{a}} - \mathbf{C}'_{\mathbf{a}}$ can be considered achievable since for any arbitrary $\epsilon > 0$, $\mathbf{a}' - \epsilon \in \mathbf{C}'_{\mathbf{a}}$.

3.3 Restrictive policies

So far we have been considering policies without any constraint on the policies other than that due to the limit on paging capacity. Now let us consider restrictive policies under which we allow any mobile to be paged in at most one cell during a slot *i.e.* blanket paging is not allowed. We shall call this a single stage policy in contrast to an unrestricted policy which is also called the multistage policy. As a result, in the logical queuing network each link emanating from a node now represents only a single cell.

Before proceeding we first illustrate the difference between single and multi-stage policies by looking at a two cell system.

Suppose a system consists of two cells 1 and 2. Then the queuing model representation of this system would consist of a set of four nodes. Further, with no constraints on the maximum possible number of cells in which a single mobile can be paged simultaneously we have a set of five links as shown in figure 3. This corresponds to the multistage policy.

Now suppose that we constrain the maximum possible number of cells in which a mobile can be simultaneously paged to one. Then the links which

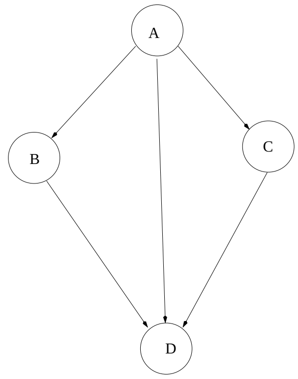

Figure 3: *Queuing model representation of multistage policies for a two cell system*

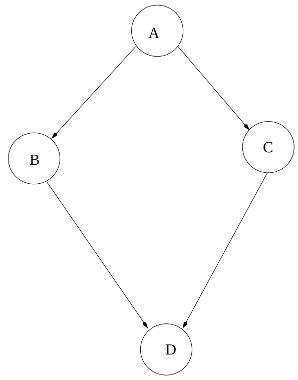

Figure 4: *Queuing model representation of single stage policies for a two cell system*

represent paging over more than one cell are not allowed in the queuing model representation of the system with constraints. Such a system is shown in figure 4. This corresponds to a single stage policy. Thus the difference between single stage and multistage policies is that the number of links allowed in the queuing model representation of a single stage policy is less than the number of links allowed in the queuing model representation of a multistage policy with the difference increasing rapidly as the number of cells in the system increases.

Note that in both these figures node A represents the set of cells 1 and 2, node B possibly represents node 1 while node C possibly represents node 2. Thus single stage policies simplify the model by eliminating many of the links considered.

Having clarified the difference between single stage and multistage policies, we next prove the main result of this section. The main result is that in order to achieve the maximum discovery rate in the system being considered, it is sufficient to consider only the single stage policies. Restating the condition A for the single stage policy, we have

$\hat{A}$. For every class $k = 1, .., K$ and set of cells S there are nonnegative numbers f^k_{Sm}, $m \in S$ such that.

$$\hat{A}1: \quad \sum_{k=1}^{K} \sum_{S} \frac{f^k_{Sm}}{p^k_S(S-m)} \leq l_m \tag{6}$$

and at the non-entry cells we have

$$\hat{A}2: \quad \sum_{m \notin S} f^k_{(S\cup\{m\})m} = \sum_{m \in S} \frac{f^k_{Sm}}{p^k_S(S-m)} \tag{7}$$

while at the entry cells given a discovery rate vector $\mathbf{a}$ we have

$$\hat{A}2: \quad a_k + \sum_{m \notin S} f^k_{(S\cup\{m\})m} = \sum_{m \in S} \frac{f^k_{Sm}}{p^k_S(S-m)} \tag{8}$$

Next, we show that, whenever constraints A1 and A2 for the multistage policies are satisfied, there exist flow vectors such that the corresponding constraints for the single stage policies are satisfied as well. But before that we give an expression for flow vectors in case of the restricted policies given flow vectors for the unrestricted policies.

Given two vectors a and b not necessarily of the same length, we say that, a dominates b, denoted as $a \succ b$ if $i \in a > j \in b \; \forall j \in b, \forall i \in a$. Now we define

$$\begin{aligned} \bar{f}^k_{Sm} = \quad & p^k_S(S-m) \sum_{(S'':m\in S''\subset S)} \sum_{S':S\cap S'=\emptyset, S''\succ S'} \frac{f^k_{(S\cup S')(S'\cup S'')}}{p^k_{S\cup S'}(S-S'')} \\ & \prod_{k=1}^{n'} p_{(S\cup S'-\sum_{i=1}^{k-1} m_i)}(S\cup S' - \sum_{i=1}^{k} m_i) \end{aligned} \tag{9}$$

where the set of cells S' is arranged as $\{m_{S'1}, m_{S'2}, \cdots, m_{S'n'}\}$ with $m_{S'i} < m_{S'j}$for $i < j$ and $n' = |S'|$.

As can be seen $\bar{f}^k_{Sm}$ can be interpreted as representing the flow rates in a restricted single stage policy. Note that the numbers $f^k_{SS'}$ on the right of (9) are all well defined. The main motivation for the above definition follows from the fact that we need to reallocate the multistage flows into single stage flows. This has to be done while ensuring that the reallocated flows are also feasible without destroying the system stability. The $\succ$ relation ensures that any multistage flow is reallocated in a unique manner thereby ensuring flexibility while the multiplication by the probability terms ensures that stability is also maintained.

Lemma 3 *Let condition A be satisfied for some vector* $\mathbf{a}$. *Then there exist numbers* $\bar{f}^k_{\bar{S}m}$, $m \in S$ *such that condition* $\hat{A}$ *is also satisfied for the paging request generation rate vector* $\mathbf{a}$.

Proof:

Since constraint A1 is satisfied, there exist numbers $f^k_{SS'}$, $S' \subset S$ such that

$$\sum_{k=1}^{K} \sum_{S} \sum_{S':m\in S'} \frac{f^k_{SS'}}{p^k_S(S-S')} \leq l_m \tag{10}$$

Partition S' into $\bar{S}'$ and $\bar{S}''$ where $\bar{S}'$ contains all the cells with the cell identification number strictly less than m and $\bar{S}''$ contains the cell m and the other cells with identifiers greater than m. Further, partition S as $S = \bar{S} \cup \bar{S}'$. Note that, by construction we have $\bar{S}' \prec \bar{S}''$. Also $m \in \bar{S}'' \subset S' \subset S$.

$$\begin{aligned}
\sum_{k=1}^{K} \sum_{S} \sum_{S':m\in S'} \frac{f^k_{SS'}}{p^k_S(S-S')} &= \sum_{k=1}^{K} \sum_{S} \sum_{S':m\in S'} \frac{f^k_{(\bar{S}\cup\bar{S}')(\bar{S}'\cup\bar{S}'')}}{p_{(\bar{S}\cup\bar{S}')}(\bar{S}-\bar{S}'')} \\
&= \sum_{k=1}^{K} \sum_{\bar{S}} \sum_{\bar{S}':m\in S'} \sum_{\bar{S}''} \frac{f^k_{(\bar{S}\cup\bar{S}')(\bar{S}'\cup\bar{S}'')}}{p_{(\bar{S}\cup\bar{S}')}(\bar{S}-\bar{S}'')} \\
&> \sum_{k=1}^{K} \sum_{\bar{S}} \frac{\bar{f}^k_{\bar{S}m}}{p^k_{\bar{S}}(\bar{S}-m)}
\end{aligned}$$

Hence $\hat{A}1$ is also satisfied.

Now we show that using the same flow rates $\bar{f}^k_{\bar{S}m}$ as above constraint $\hat{A}2$ is also satisfied given that A2 is satisfied.

$$\begin{aligned}
&\sum_{m\notin S} \bar{f}^k_{(S\cup\{m\})m} = \sum_{m\notin S} \bar{f}^k_{(\hat{S})m} \text{ with } S\cup m = \hat{S} \\
&= \sum_{m\notin S} \sum_{S''} \sum_{S'} \frac{f^k_{(\hat{S}\cup S')(S'\cup S'')}}{p^k_{\hat{S}\cup S'}(\hat{S}-S'')} \prod_{k=1}^{n'} p_{\hat{S}\cup S' - \sum_{i=1}^{k-1} m_i}(S\cup S' - \sum_{i=1}^{k} m_i) p^k_{\hat{S}}(\hat{S}-m)
\end{aligned}$$

$$= \sum_{m \notin S} \sum_{S'} \frac{f^k_{(\hat{S} \cup S')(S' \cup m)}}{p^k_{\hat{S} \cup S'}(\hat{S} - m)} \prod_{k=1}^{n'} p_{(\hat{S} \cup S' - \sum_{i=1}^{k-1} m_i)}(\hat{S} \cup S' - \sum_{i=1}^{k} m_i) p^k_{\hat{S}}(\hat{S} - m) +$$

$$\sum_{m \notin S} \sum_{S'' : m \neq S''} \sum_{S'} \frac{f^k_{(\hat{S} \cup S')(S' \cup S'')}}{p^k_{S \cup S'}(S - S'')} \prod_{k=1}^{n'} p_{S \cup S' - \sum_{i=1}^{k-1} m_i}(S \cup S' - \sum_{i=1}^{k} m_i) p^k_{\hat{S}}(\hat{S} - m)$$

Now let $\bar{S}' = S' \cup m$ where $m \notin S$. Hence, using the fact that A2 holds, we have

$$\begin{aligned}
\sum_{m \notin \bar{S}} \bar{f}^k_{(\bar{S} \cup \{m\})m} &= \sum_{\bar{S}' \cap S = \emptyset} f^k_{(S \cup \{\bar{S}'\})\bar{S}'} + \\
\sum_{m \notin S} \sum_{S'' : m \neq S''} \sum_{S'} \frac{f^k_{(\hat{S} \cup S')(S' \cup S'')}}{p^k_{\hat{S} \cup S'}(\hat{S} - S'')} & \prod_{k=1}^{n'} p_{(\hat{S} \cup S' - \sum_{i=1}^{k-1} m_i)}(\hat{S} \cup S' - \sum_{i=1}^{k} m_i) p^k_{\hat{S}}(\hat{S} - m) \\
&= \sum_{S' \subset S} \frac{f^k_{SS'}}{p^k_S(S - S')} + \\
\sum_{S'' : m \notin S''} \sum_{\bar{S}'} \frac{f^k_{(S \cup \bar{S}')(\bar{S}' \cup S'')}}{p^k_{S \cup \bar{S}'}(S - S'')} & \prod_{k=1}^{n'} p_{S \cup \bar{S}' - \sum_{i=1}^{k-1} m_i}(S \cup \bar{S}' - \sum_{i=1}^{k} m_i) \\
&= \sum_{m \in S} \frac{\bar{f}^k_{Sm}}{p^k_S(S - m)}
\end{aligned}$$

◇

Hence, any paging request generation rate vector which is achievable by any multi-stage paging policy is also achievable when the paging policy is restricted as above to be a single stage policy. As a result, if the goal is to attain the maximum discovery rate in the system, then it is enough to concentrate just on the restricted single stage paging policy whereby each mobile is paged in at most one cell during a single time slot. As a result, we have the next theorem

Theorem 2 *If the given paging request generation rate vector* **a** *is achievable using an unrestricted multistage paging policy, then it is also achievable by the class of restricted single stage policies.*

Proof:

Since the given discovery rate vector **a** is achievable under the unrestricted paging policy, condition A holds. Hence, by lemma 3, condition $\hat{A}$ also holds. Hence, it can be proven similarly as in the case of the multistage policies that **a** is indeed a discovery rate vector even when considering the class of restricted single stage policies.

◇

Note that, in fact the policy π given earlier can also be shown to be an optimal policy in terms of the maximum discovery rate while considering only the class of single stage policies. The proof given earlier holds with the only difference that, the routing matrix in this case while being defined as earlier will be different.

The implementation of the maximum throughput policy given earlier for the multistage policies suffers from two problems. The first is the large number of nodes and links to be considered. With the multistage policies for a system having N cells the number of nodes to be considered is $2^N - 1$ and the number of links much more. The second problem is that of solving an optimization problem at stage 2 which is also quite cumbersome. The single stage optimal policy helps in alleviating these problems to some extent but still suffers from the curse of dimensionality leading to implementation complexity.

Next we include an example demonstrating the calculation of the boundary of the discovery rate vectors for a system with 2 cells. We later also specify an optimal paging algorithm assuming two cells and any number of classes.

Example 1: Consider a system with 2 cells and one class, *i.e* $N = 2$ and $K = 1$. We would like to calculate the value of the maximum possible paging request generation rate vector for which the system would be stable.

Considering the single-stage policies we use constraints $\hat{A}1$ and $\hat{A}2$. Writing down the corresponding equations, we have a total of 5 equations and 5 unknowns which could be solved for the maximum paging request generation rate vector which is given as

$$\mathbf{a} = \frac{l_1(1 - p_{12}(2)) + l_2(1 - p_{12}(1))}{1 - p_{12}(1)p_{12}(2)} \tag{11}$$

Note that in the above we use the weak inequality in case of $\hat{A}1$ and thereby obtain the boundary of the achievable discovery rate vectors. It should also be pointed out that as the number of classes and number of cells increases it is not possible to obtain a closed form expression for the maximum achievable paging request generation rate vectors.

In the next section, we provide an implementable algorithm for paging assuming two cells in the system and show that this achieves the maximal discovery rate for the case with two classes.

4 SIMULATION RESULTS

We assume that the system has two cells and K classes. The statistics of mobility pattern is assumed to be known in the form of the location probability of mobiles belonging to different classes in the two cells. For such a scenario we next give an algorithm which attains the maximum discovery rate.

The algorithm is

1. We assume that there are $K+2$ queues. Queue 0 contains the requests which have a zero probability of being found in cell 2. The probability of being found in cell 1 decreases while going from queue 1 to queue K. Queue $K+1$ contains those requests which have a zero probability of being found in cell 1. Thus queues 1 to K correspond to the classes 1 to K if the classes have been ordered such that $p_{12}^{i}(1) >= p_{12}^{j}(1)$ for all $i \leq j$.

2. When a request of class i arrives it is directed to the corresponding queue i.

3. To decide the mobiles to be paged in cell 1, the paging controller starts from queue 0 and goes all the way up to queue N selecting paging requests to be paged over each of its paging channels until either there are no more paging requests in these queues or the number of paging channels in cell 1 is exhausted.

4. To decide the mobiles to be paged in cell 2, the paging controller starts from queue $K+1$ and goes all the way up to queue 1 selecting paging requests until either there are no more paging requests in these queues or the number of paging channels is exhausted.

5. When the mobiles which are paged are found, the request is deleted. If the mobile has been paged in cell 1 and not found the request is transferred to queue $K+1$. Similarly if the mobile was paged in cell 2 and not found the request is transferred to queue 0. Note that requests from queue 0 or queue $K+1$ which are paged are found with probability 1. We go back to step 2 above.

Next, we specify the achievable paging request generation rate based on the above algorithm for the case of 2 cells and K classes. Using the earlier approach it would not be possible to obtain a closed form expression. Further, in case of 2 classes the given expression is the same as the one given in the earlier section by equation 11.

Consider a system with $N = 2$ cells and K classes. We would like to calculate the value of the maximum possible paging request generation rate vector for which the system would be stable.

In this case, since we are interested in the maximal achievable rate, we assume that only mobiles of class 1 or class K enter the system. This is because mobiles corresponding to these classes have the highest probability of being in cell 1 and cell 2 respectively. Then the paging channels in cell 1 have to serve only queues 0 and 1 while paging channels in cell 2 have to serve only queues K and $K+1$. Let x_0 be the proportion of paging channels in cell 1 serving queue 0 in steady state. Similarly let y_0 be the proportion of paging channels in cell 1 serving queue $K+1$ in steady state. Then for stability we have

$$\begin{aligned} a_1 &= (1-x_0)l_1 \\ a_2 &= (1-y_0)l_2 \\ x_0 l_1 &= p_{12}^2(1)(1-y_0)l_2 \qquad (12) \\ y_0 l_2 &= p_{12}^1(2)(1-x_0)l_1 \qquad (13) \end{aligned}$$

From the above equations we obtain the maximal achievable rate a as

$$\begin{aligned} a &= a_1 + a_2 \\ &= \frac{l_1(1-p_{12}^1(2)) + l_2(1-p_{12}^2(1))}{1-p_{12}^2(1)p_{12}^1(2)} \qquad (14) \end{aligned}$$

We next provide results of simulating the above algorithm. We assume that there is one paging channel in each cell. The paging requests generation rate vector is assumed to be Poisson. For both the simulation sets next we simulate the system for 5000 time units and average out the results over five different seeds for the random number generator.

4.1 Simulation Set I

We assume that there are two classes in the system. Mobiles of class 1 are assumed to have a probability of 0.8 of being present in cell 1 while mobiles of class 2 are assumed to have a probability of 0.4 of being present in cell 1. We let the paging request generation rates of mobiles of both class 1 and class 2 range from 0.00002 to 2.0 in steps of 0.1. The very small value of 0.00002 (reciprocal of the time of simulation) is taken to simulate the system when only one class of mobiles is coming in. Figure 5 shows the average number of mobiles found per slot for each of the two classes under this scenario. The boundary of the discovery rate vectors gives the paging request generation rate region for which the system can be stabilized under the given paging policy. If only requests for mobiles of class 1 were coming in then the achievable rate would be 1.19 as obtained from equation 11. Similarly, the value for the achievable rate of mobiles of class 2 would be 1.316. We see from the figure that these values are achieved when paging request generation requests are assumed to belong to only one class.

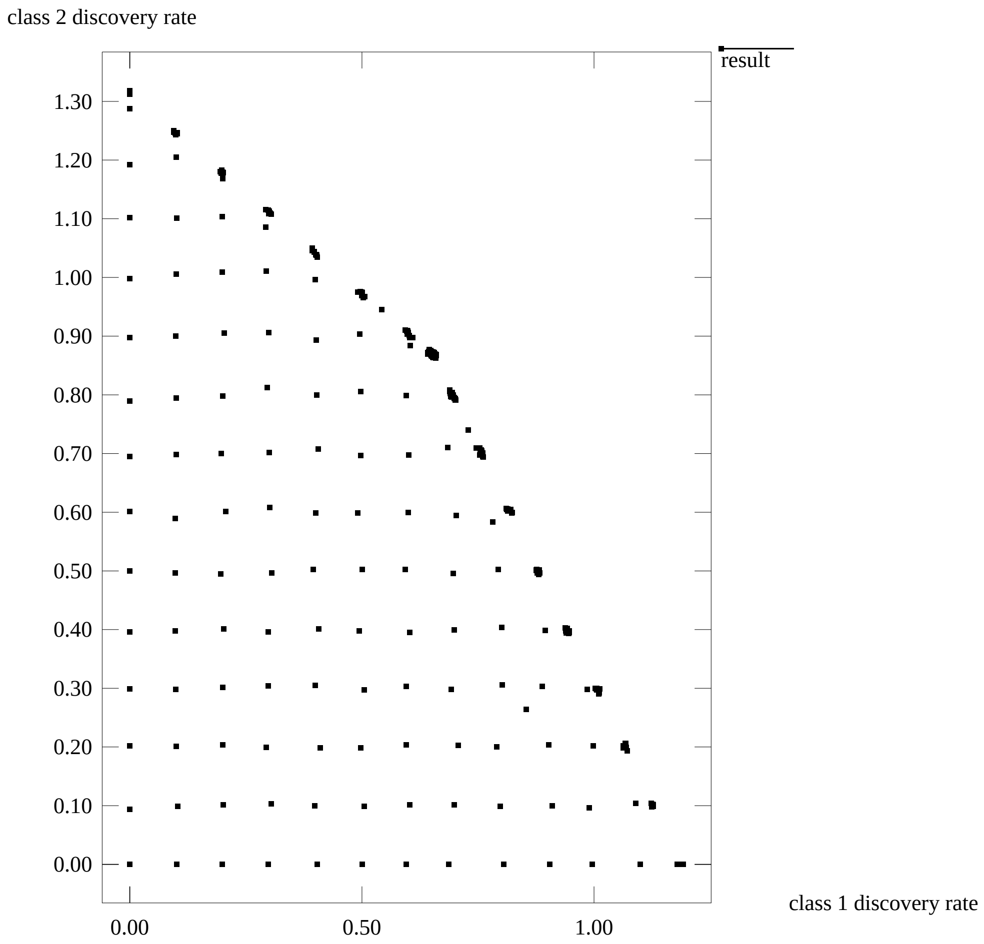

Figure 5: *Achievable paging request generation rate region for two classes*

4.2 Simulation Set II

In the next case we assume that there are more than two classes of mobiles. In this case we look at the paging request generation rate for all the classes of mobiles. Results showing the discovery rate for 4, 5 and 10 classes of mobiles are as shown in figure. The percentage of mobiles found as a function of the overall paging request generation rate for 4, 5 and 10 classes is also shown in figure. The probability of a mobile of class i being present is cell 1 assuming K classes is assumed to be $\frac{K+1-i}{K+1}$. Figure 6 shows the discovery rate as a function of the paging request generation rates of all mobiles. Figure 7 shows the ratio of mobiles found over the total requests into the system as a function of the Poisson arrival rate. Note that in both the figures, the system is stable in the region where the graphs are straight lines.

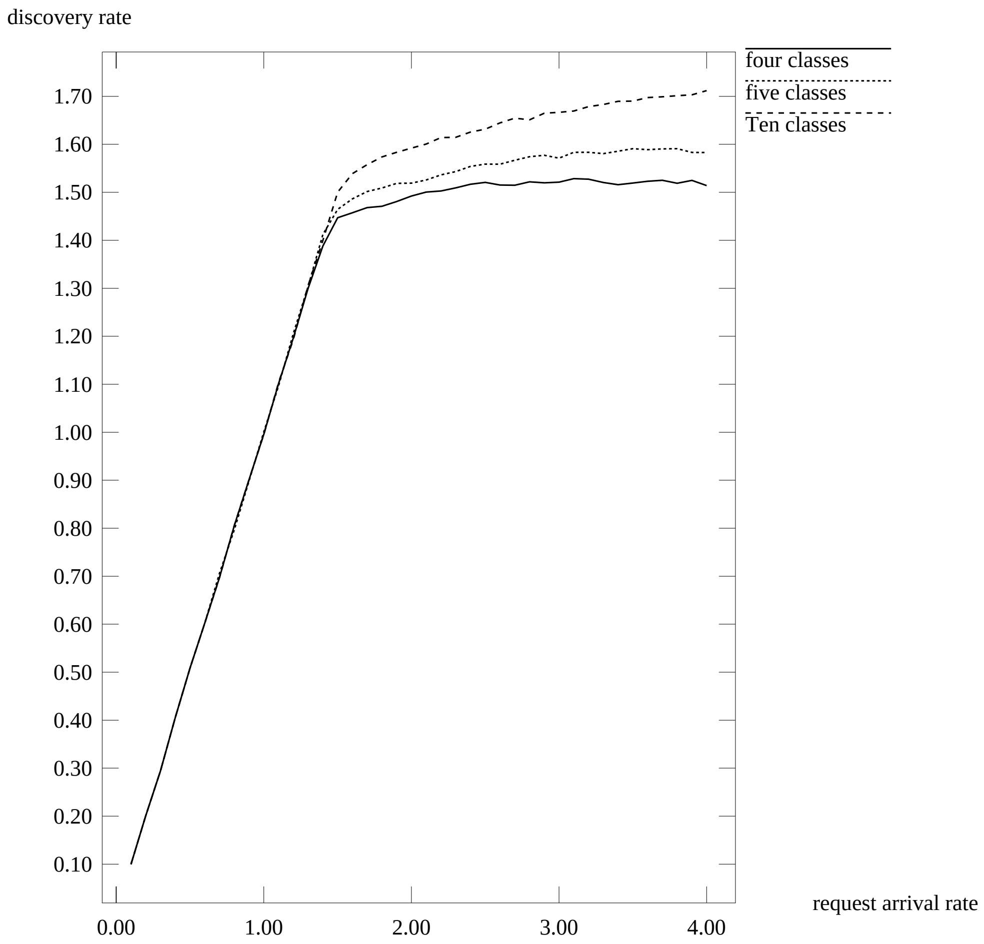

Figure 6: *Discovery rate vector behavior for two cell system*

5 CONCLUSION

In this paper, we have considered a wireless network from the point of view of achieving the maximum discovery rate possible in the system. The wireless network is modeled logically as a queuing network. One of the main contributions of this work is the characterisation of the region of the discovery rate vectors. A maximum throughput policy for the wireless network has also been specified. Another important result shown is that, in order to achieve maximum discovery rate, it is enough to consider the simpler and less costly single stage policies.

An example of a two-cell system is illustrated to demonstrate how the given results help in calculating and achieving the maximum achievable discovery rates. But for large systems problems do remain with regards to the efficient determination of whether a given discovery rate vector belongs to the set $\bar{C}$ since the answer is highly system dependent. An algorithm for this purpose is the subject of future research.

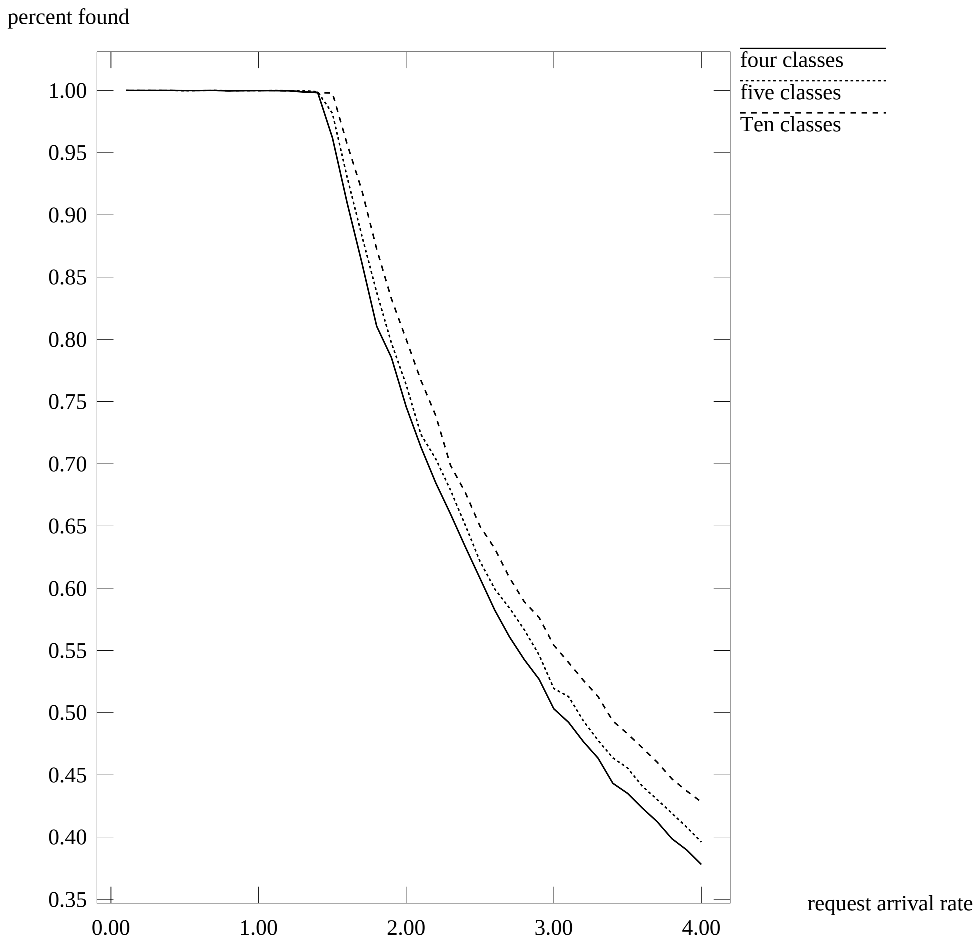

Figure 7: *Percentage of mobiles found as a function of paging request generation rate for two cell system*

References

[1] I.F. Akyildiz and J.S.M. Ho. Dynamic mobile user location update for wireless PCS networks. *Wireless Networks*, 1:187–196, 1995.

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

[3] D.J. Goodman, P. Krishnan, and B. Sugla. Minimizing queuing delays and number of messages in mobile phone location. *Mobile Networks and Applications*, 1(1):39–49, August 1996.

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

[5] G. Pollini and S. Tabbane. The intelligent network signaling and switching cost of an alternative location strategy using memory. *Proc. IEEE Vehicular Technology Conference*, May 1993.

[6] Rezaiifar R. and Makowski A. M. From optimal search theory to sequential paging in cellular networks. *IEEE JSAC*, 15(7):1253–1264, Sept 1997.

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

[8] C. Rose and R. Yates. Ensemble polling strategies for increased paging capacity in mobile communication networks. *ACM Journal of Wireless Networks*, 3(2):159–167, May 1997.

[9] L. Tassiulas and F.M. Anjum. A hierarchical multi-resolution registration structure for mobility tracking. IEEE Conf. on UPC, Boston, Sep 1996.

[10] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control*, 37(12):1936–1948, December 1992.

[11] T. Woo, T.F. LaPorta, J. Golestani, and N. Agarwal. Update and search algorithms for wireless two way messaging: Design and performance. In *Infocom*, San Francisco, USA, Mar 31-Apr2 1998.

[12] Michalas Y. and L. Tassiulas. Multi-user paging policies in cellular wireless networks. In *MMT98*, George Washington University, October 1998.