Optimization Based Bandwidth Allocation in Mobile Cellular Networks

Samrat Ganguly and Badri Nath
Dataman Lab, Dept of Computer Science
Rutgers University, NJ, USA
{ganguly,badri}@cs.rutgers.edu

Abstract—**Efficient bandwidth allocation strategy with simultaneous fulfillment of QoS requirement of a user in a mobile cellular network is still a critical and an important practical issue. We explore the problem of minimizing the amount of time for which bandwidth has to be allocated in a cell while meeting the QoS constraint. With the knowledge about the the arrival and residence time distribution of a user in a cell, the above problem can be optimally solved using a dynamic programming based approach in polynomial time. To be able to use the solution, we provide a mechanism for constructing the arrival/residence time distribution based on the measurement of hand-off events in a cell. The above solution allows us to propose an optimal time based bandwidth reservation and call admission scheme. By being scalable and distributed, the proposed scheme justifies for practical implementation. Simulations results are also presented to show the effectiveness of the scheme to achieve the target QoS level and optimal bandwidth utilization.**

I. Introduction

The new upcoming wireless infrastructures such as 3G and 4G are deemed to support broad band data applications and new services. The expected services will also include multimedia applications that need real time guarantees. To meet the above applications the service providers ought to adopt some form of a reservation scheme or service differentiation to support high quality of service and at the same time extract high utilization from the network resources.

In a cellular network, a mobile user may visit different cells in his lifetime. In each of these cells, resources must be available to support the mobile user else the user will suffer a forced termination of his call in progress. Therefore, carefull resource allocation along with call admission control is required to mitigate the chances of forced termination or dropping of a call. Due to the uncertainty imposed by the mobility of the user, it is considered impractical from the utilization stand point to completely eliminate the chances of dropping a call. Thus keeping the probability of a user getting dropped (P_{drop}) below a pre-specified target value is considered as a practical design goal for any resource allocation scheme. Achieving the above goal provides the probablistic quality of service (QoS) guarantee as desired by a mobile user. On the contrary, from a network providers stand point , with a fixed given cell capacity, the objective is to extract high utilization by minimizing the overall resources allocated for a user. In a reservation based framework, the overall resources allocated per user has two principal components: the spatial resources and the temporal resources. Minimizing the spatial resources requires reducing the number of cells where bandwidth needs to be reserved and can be done based on considering either apriori knowledge or prediction about users future movement pattern. Based on this consideration, several schemes have been proposed that uses mobility profile [1], [2], direction prediction [3], knowledge about possible geographic routes with the help of ITS Navigation system [4] etc.

Temporal resources on the other hand requires minimizing the amount of time the resources are reserved in a cell. Although in future, it may be possible for a user to provide exact information about the cells he is likely to visit, it may be still difficult for the same user to provide apriori information about when he may visit these cells and how long he is going to stay in each cell. Consequently, with the uncertainty about the temporal aspects in users mobility behaviour, it becomes challenging task to minimize the time for which bandwidth reservation must be held in the cells for the user. Our focus here is to explore the use of time aspects in users mobility towards minimizing the temporal resources allocated for a user subject to meeting QoS constraint on drop probability.

Prior work in this area was based purely on call admission control without keeping any reservation states. These schemes such as in [5], [6], [7], [8], [9] were often based on either dynamical or statical prediction of the steady state distribution of users' demand in different cells. In contrast, several schemes based on keeping reservation states and per user monitoring were proposed in [1], [3], [10], [11] and found to perform better than the above schemes based on simulation experiments presented in [12]. Some of these reservation based scheme such as in [1] were just based on estimating spatial per user resource

demand while others as in [3], [10], [11] also included the time aspects in users' demands in their scheme.

It is worth mentioning that most of the allocations schemes were based on predicting per/aggregate user demand and employing it to provide QoS through call admission control with/without reservation states. However, the problem of minimizing the allocated resources to meet the drop probability constraint has not been considered in the existing schemes. In essence, majority of the allocation schemes are parametric in nature in the sense that these schemes provide a parameter which can be used to obtain a particular level of QoS(drop probability) while trading-off utilization.

The main contribution of our work is to develop an optimal scheme for resource allocation that minimizes the amount of time resources are reserved in a cell. In order to do so, we cast the resource minimization problem meeting the drop probability as a optimization problem and adopt a dynamic programming based approach to solve the problem optimally in polynomial time. Our solution to this problem only needs to know the arrival and residence time probability distribution of users in a given cell based on the a very general assumption that the above distribution follows a stationary stochastic process (a necessary condition for predicting resource demand under any circumstances). Finally, to apply the solution for practical situations, we develop a scheme for constructing the arrival/residence time distribution based on the measurements of hand-off events and propose a time based reservation framework to enable the optimal resource allocation. Our proposed scheme is scalable by not keeping any per user states and does not rely on remote cell query or messaging except at the time of reservation set-up.

The rest of the paper is organized as follows. In section 2, we formulate the optimization problem and present the algorithm for obtaining the solution in section 3. We present a scheme for constructing the arrival/residence time distribution in section 4. We present our proposed time-based reservation reservation framework in section 5. Simulation results are presented in section 6. Finally the the main conclusion is drawn in section 7.

II. Problem Statement

In this section, first we describe the cellular system with bandwidth reservation along with terminologies we use in the problem formulation. Subsequently, we show how we can formulate the optimization problem with the objective of minimizing the time frame for which bandwidth is reserved in a cell to meet the constraint on the drop probability.

A. System Description and Terminologies

In a cellular network, let cell i be the current location of the mobile user. Let us consider a single cell j which the mobile user may visit in his lifetime. Therefore bandwidth must be allocated and reserved in cell j. Our concern is about how the bandwidth allocated must be reserved over time in cell j. For simplicity, let us assume that the time is divided into integer slots and that bandwidth is reserved on slot basis in a given cell. The current time when the reservation is set-up corresponds to slot 0. Let *L* denote a set of slots where bandwidth is reserved for the mobile user. For example $L = \{1, 2, 4\}$ represents a scenario where bandwidth is reserved in slots 1,2 and 4 for a particular user. Let t_d be the time when the user departs a given cell either due to call termination or due to hand-off to an adjacent cell. Therefore, the maximum value of t_d corresponds to the maximum value of slot index l_{max} that the set *L* can have since there is no point in reserving bandwidth when the user is not present in the cell. Keeping this in mind, we can safely say that the bandwidth maybe reserved in the slot index l where $0 \leq l \leq l_{max}$. In other words, the range of slots $[0 \ldots l_{max}]$ corresponds to the maximum time window the user can stay in the given cell j.

Let us now define the random variable $\mathbf{X}_a = k$ of lattice type as the outcome that a user has arrived at the k^{th} slot in a given cell j. Given the statistics of $\mathbf{X}_a$, we can define $f_a(X_a)$ as the corresponding arrival time probability density function(*pdf*). $f_a(X_a)$ is a discrete function with the property that $\sum_0^{l_{max}} f_a(X_a) = 1$. Similarly, we define the random variable $\mathbf{X}_r = n$ as outcome that a user departs the cell j at the n^{th} time slot. $f_r(X_r)$ thus denotes the corresponding discrete residence time *pdf* for $\mathbf{X}_r$. At this point we assume that both the above density functions are known to us about a user and cell j. In the next section we discuss how to contruct the density functions.

B. Problem Overview

Figure 1 shows a mobile user(mh) who is expected to visit cell j. Suppose he makes his reservation at time t_i and let the call departure time be t_d. In such a case, bandwidth needs to be allocated in time slots($0 \ldots l_{max}$) as shown in Fig. 1. But since the exact time of arrival and departure at cell j is not known apriori, therefore it becomes a question as to which slots bandwidth should be allocated to meet drop probability requirement of the user. Since allocating bandwidth in all slots results in lower utilization, therefore, our objective is to find the minimum number of slots where bandwidth needs to be allocated subject to meeting the constraint on the drop probability.

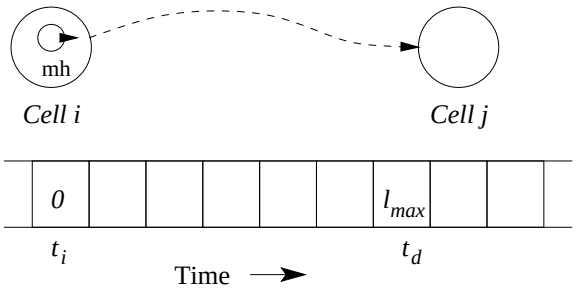

Fig. 1. Bandwidth reservation at cell j in time

C. The Optimization Problem

In order to provide a certain bound on the drop probability to a given user during his visit to cell j, one must allocate bandwidth over the time slots. Our goal here is to minimize the number of slots where the bandwidth must be reserved to meet the constraint on the drop probability. Therefore, we need to relate the drop probability to the slots where bandwidth is reserved for the user. In order to do so, let us first define a vector $B[0 \dots N]$ where the element $B[i]$ can be 0 or 1 and N refers to l_{max}. $B[i] = 1$ means resources are reserved for the given user in the i^{th} slot and $B[i] = 0$ means otherwise. Based on this definition, we can define L as $\{i|B[i] = 1\}$. For a given B, we also define a new vector $P[0 \dots N]$ where $P[i]$ is defined as follows.

$$P[i] = 0 \quad \text{IF} \quad B[i] = 0$$

$$P[i] = k \quad \text{IF} \quad \forall\, j = i \dots i + k - 1,\ B[j] = 1$$

Therefore, $P[i]$ basically denotes the number of consecutive *1's* starting from the i^{th} position in B. In that case, if a user arrives at cell j in the i^{th} slot and $P[i] = k$, it implies that the user will find resources reserved for him for the next k slots. If this user stays beyond k slots, he may be dropped. Therefore, the maximum conditional drop probability under the condition that the user arrives at i^{th} slot is given by $P_{cdrop}(i) = 1 - F_r(P[i])$ where $F_r(\cdot)$ is residence time distribution function[1]. For example, if no bandwidth is reserved in the i^{th} slot($P[i] = 0$) and since $F_r(P[i]) = 0$, $P_{cdrop}(i)$ becomes equal to 1. Thus the total drop probability for a given user will be given by

$$P_{drop}(B) = \sum_{i=0}^{N} f_a(i) * P_{cdrop}(i). \tag{1}$$

We observe that the drop probability depends upon B and the total resources allocated for a user is given by $\sum_{i=0}^{N} B[i]$. Our aim is to minimize the amount of resources used to provide a given QoS defined by the maximum drop probability. We therefore specify our optimization problem as follows.

$$Find\ B\ s.t.$$

$$\sum_{i=0}^{N} B[i]\ is\ minimized$$

$$P_{drop}(B) < T$$

where $T \in [0 \dots 1]$ is the prespecified upper bound on the drop probability corresponding to a given level of QoS.

III. Algorithm for finding optimal B

Finding B in order to minimize the sum of 1's in B subject to meeting the constraint is a combinatorial optimization problem. We use a dynamic programming based approach to devise a polynomial time algorithm in finding the optimal solution. Initially we consider $B[i] = 1$ for all i, which results in $P_{drop}(B) = 0$. Therefore inserting zeros in B may increase $P_{drop}(B)$. Our intention is to insert maximum number of zeros while keeping $P_{drop}(B) < T$. We present an iterative algorithm where in each iteration step we insert a single zero in B and we stop at the iteration step where the constraint is no more satisfied or $P_{drop}(B) \geq T$. We denote the updated B at the end of the k^{th} iteration step as B_k which has k zeros. The solution of B_k at the end of the k^{th} iteration step in the algorithm provides the position of the 0's in B_k for which $P_{drop}(B_k)$ is a minimum with k zeros.Therefore, in the k^{th} iteration we are trying to find out which combinations of k zeros in B gives the minimum drop probability. Consequently, if we stop at the i^{th} iteration step, we claim that (i-1) zeros in the optimal permutation (found at the end of $(i-1)^{th}$ iteration) gives the optimal B. The above approach is based on the following proposition which we use in the algorithm.

Proposition 1: If the i^{th} position of B has zero then

$$\begin{aligned} P_{drop}(B) &= P_{drop}(B[0 \dots i-1]) \\ &+ f_a(i) + P_{drop}(B[i+1 \dots N]) \end{aligned} \tag{2}$$

Proof: From eqn(1), we can express $P_{drop}(B)$ as follows:

$$\begin{aligned} P_{drop}(B) &= \underbrace{\sum_{j=0}^{i-1} f_a(j) \cdot P_{cdrop}(s,j)}_{A} \\ &+ \underbrace{f_a(i) \cdot P_{cdrop}(s,i)}_{B} + \underbrace{\sum_{j=i+1}^{N} f_a(j) \cdot P_{cdrop}(s,j)}_{C} \end{aligned} \tag{3}$$

[1] In cases where the base station can relinquish unreserved bandwidth for user staying beyond k slots, the drop probability will be lower than $P_{cdrop}(i)$ as defined. Here we consider the constraint on the upper bound on drop probability that serves as a QoS metric

Since $P[j]$ and hence $P_{cdrop}(j)$ in A do not depend upon the values in position $i \ldots N$ of B because of the zero in the ith position, therefore A equals to $P_{drop}(B[0 \ldots i-1])$. Also since $P[j]$ is based on values of B in the forward $(\geq j)$ positions and therefore, C is equal to $P_{drop}(B[i+1 \ldots N])$. Finally, since $P[i] = 0$ which implies $P_{cdrop}(i) = 1$ thus making B being equal to $f_a(i)$, we show that the above proposition is true. ■

We next provide the iteration steps in our proposed optimal algorithm. Let us consider N subarrays $A_0^0 \ldots A_N^0$ where A_i^0 is $B_0[i \ldots N]$. At the end of each iteration we construct a new N subarrays i.e. at the end of the k_{th} iteration we construct $A_0^k \ldots A_N^k$. We also define $POS(A_i^k)$ to be a set denoting the position of zeros in A_i^k. Initially $POS(A_i^0) = \{\emptyset\}\ \forall i$.

Iteration 1: Consider a particular subarray A_i^1 which initially has all 1's in it. We find out the position p where by inserting a zero gives a minimum value of $P_{drop}(A_i^1)$. We then update A_i^1 by inserting a zero in the pth position. We also get $POS(A_i^1) = POS(A_i^0) \cup p$. At the end of this iteration we note that $POS(A_0^1)$ gives the position of the single zero in B_0 for which the drop rate is a minimum. Therefore, we assign $B_1 = A_0^1$. We go into the second iteration if $P_{drop}(B_1) < k$.

Iteration 2: Consider a particular subarray A_i^2 which initially has all ones in it. In this iteration our intention is to find out the position of 2 zeros to be inserted in A_i^2 which gives a minimum $P_{drop}(A_i^2)$. Consider the case where the 1^{st} zero is in the p^{th} position in A_i^2, then $P_{drop}(A_i^2) = P_{drop}(A_i^2[0 \ldots p-1]) + f_a(p) + P_{drop}(A_i^2[p+1 \ldots N-i])$ from (2). Since $A_i^2[0 \ldots p-1]$ has all ones and $f_a(p)$ is fixed, therefore once we set the 1^{st} zero in p^{th} position, the a minimum $P_{drop}(A_i^2)$ will correspond to the second zero in $A_i^2[p+1 \ldots N-i]$ for which $P_{drop}(A_i^2[p+1 \ldots N-i])$ is a minimum. In that case the second zero must be in position $p + POS(A_{p+a}^1)$ which we already found in the 1^{st} iteration. Thus the minimum $P_{drop}(A_i^2)$ will correspond to the 1^{st} zero at the position given by

$$l = \min_p [P_{drop}(A_i^2[0 \ldots p-1] + f_a(p) + P_{drop}(A_{p+1}^1)]$$

Therefore A_i^2 is obtained by inserting zeros at $l, l + POS(A_{l+1}^1)$ positions and one can obtain $POS(A_i^2)$ likewise. Finally we assign $B_2 = A_0^2$. Now that we have provided sufficient background about the working of the algorithm we describe the general k^{th} iteration step.

Iteration k: Consider the subarray A_i^k where we find the position of the first zero *(only if the size of $A_i^k \geq k$)* given as

$$l = \min_p [P_{drop}(A_i^k[0 \ldots p-1] + f_a(p) + P_{drop}(A_{p+1}^{k-1})]$$

We obtain $POS(A_i^k)$ as

$$POS(A_i^k) = \{l\} \cup \{y \mid y = x + l, x \in POS(A_{p+1}^{k-1})\}$$

A_i^k is then updated by inserting zeros in the positions given in $POS(A_i^k)$. Finally, we assign $B_k = A_0^k$.

Proof of Correctness: The proof of correctness for the above algorithm in finding the optimal solution is based on the following the two propositions. In the first proposition, we show that $P_{drop}(B)$ is a monotonically decreasing function of the number of zeros in B.

Proposition 2: $P_{drop}(B_k) \leq P_{drop}(B_{k+1}) \quad \forall k.$

Proof: Let us consider the first and the second zero to be in the p^{th} and q^{th} position in B_{k+1} respectively. From the above proposition, we can write $P_{drop}(B_{k+1})$ as

$$\begin{aligned} P_{drop}(B_{k+1}) = & \underbrace{P_{drop}([1 \ldots 1 \overset{p}{0} 1 \ldots \overset{q-1}{1}])}_{C1} \\ & + \underbrace{f_a(q)}_{C2} + \underbrace{P_{drop}(B_{k+1}[q+1 \ldots N])}_{C3} \end{aligned} \tag{4}$$

If $P_{drop}(B_{k+1}) < P_{drop}(B_k)$, it follows $c1 + c2 + c3 < P_{drop}(B_k)$ from the above eqn(4). Now if we insert a 1 in the p^{th} position in B_{k+1}, we get $c = P_{drop}(B_{k+1}[0 \ldots q-1]) \leq c1 + c2$, since adding the 1 can only increase $P[i]\ \forall i = 0 \ldots q-1$. Therefore we get $c + c3 < P_{drop}(B_k)$ or we constructed a B' from B_{k+1} with k zeros and $P_{drop}(B') < P_{drop}(B_k)$. But such a construction contradicts the definition of B_k and hence proves the proposition. ■

Proposition 3: B_k found by the above algorithm is what it's supposed to be: that is, $B = B_k$ achieves minimum of $P_{drop}(B)$.

Proof: By induction, on the size of the array B, the base case is easy. Now assume that the algorithm finds B_k correctly for all $k \leq N$ for all inputs. Equation (2) plays the crucial role in the induction step. If $k = l_{max}$ then there is nothing to prove. So assume that $k < l_{max}$, in that case there is some index i with ith entry 0. In the algorithm, we try all the l_{max} possible values of i; and for a given value of i we get *a set of* two *independent* subproblems: For all j, k such that $j + k = n$, find B_j for the left subarray (upto index $i-1$), and find the B_k for the right subarray (from index $i+1$ to n).

This takes care of all the possible ways in which B_{n+1} could occur, and since the algorithm tries them all, it finds the minimum value.

■

Monotonicity and the above proposition imply that B_k is the optimal B (optimal for the optimization problem

above) if we stop at the $(k+1)$th iteration of the algorithm.

Complexity: First we discuss the space complexity needed by the above algorithm. We note that at the i_{th} iteration step we need to store the value of P_{drop} and POS for all subarrays obtained in the $(i-1)^{th}$ iteration step. The above storage needs $O(N)$ space for storing P_{drop} values and $O(N^2)$ space for storing POS values. Now for time complexity we note that to find the position of the first zero in A_i^k, it takes $O(N^2)$ time to find $A_i^k[0 \ldots p-1]$ for all p and $(N-i)$ time to find the minimum giving a total of $O(N^2 * (N-i))$ time. Therefore, to find the first zeros for all the subarrays takes $O(N^4)$ time. The other operations in the iteration takes O(1) time. Since $A_i^k[0 \ldots p-1]$ always has all ones in it and we anyway evaluate it in the first iteration and therefore do not need to evaluate it again in subsequent iterations if we store the values of it. In that case the time complexity of a single iteration reduces to $O(N^2)$. Finally, given that we stop at ith iteration the total time complexity of the scheme becomes $O(i * N^2)$.

It should also be noted that we presented the above algorithm for bandwidth optimization in a single cell for the sake of distributed implementation. The solution does not preclude the scenario where optimization needs to be done over all cells user may visit. In that case one needs to consider a cluster of cells conceptually equivalent to a single cell and extend the application of the solution.

IV. Arrival/Residence time *pdf*

The optimal allocation algorithm discussed in the last section requires construction of the arrival and residence time *pdf* $f_r(\cdot)$ and $f_r(\cdot)$ respectively. From the definition of the arrival time *pdf* in section 2, we see that it refers to the probablity distribution of the *time* a user takes to reach cell i from his current location at cell s. Therefore, users residing in a different cell s' will have a different arrival time *pdf* to cell i. For that reason, our sample space for constructing the arrival time *pdf* can be only restricted to the information about past users originating at cell s and visiting cell i. Consequently, a base station for the need of resource allocation, requires to know the arrival time *pdf* for each call originating cell. Residence time *pdf*, on the other hand, can be based on the sample space of past users residence time in cell i. Our justification for using past history in constructing the arrival/residence time *pdf* is based on the following observation. The observation is that in a given cell a user has very low probability of acting differently from other users in the same cell. For example, inside a mall, users moving in slow walking pace, a particular user may have maximum a running pace but not a velocity of when he is driving a car in highway. In other words locality imposes on users a statistical distribution on velocity, residence time, direction etc. Next we discuss in detail how we construct the arrival/residence time *pdf*.

A. *Construction of Arrival and Residence time* pdf

The following construction of the density function is based only on monitoring the handoff events in a given cell and does not involve any remote cell query or any monitoring of per-user profile/status. For clarity sake, let us refer the arrival time *pdf* for user from the originating cell s by $f_a^s()$ and discuss the construction of $f_a^s()$ at cell i.

Consider a user that arrives at cell i, at handoff the user informs the base station(*bs*) of cell i with his call orginating time t_i and the originating cell. To obtain $f_a^s()$, the *bs* consider users only from cell s and based on current time t finds the slot where he has arrived. The specific slot is found by mapping the value $\Delta t = t - t_i$ to integer slots[2]. Therefore the arrival event $(X_a^s = k)$ denotes that a user from cell s has arrived in the current cell i at slot k. We now consider a window of time W and find the relative frequency of each event based on its occurrence on the time window W. Therefore, at current time t, we look back up to t-W time to find out how many times the event $(X_a^s = k)$ has occurred and let it be N_k. Let N be the total number users who came from cell s during the time interval [t-W : t]. We obtain the relative frequency of the event $(X_a^s = k)$ as N_k/N. Analogously, we obtain the relative frequency for all other event for $k = 0, \ldots, l_{max}$ and construct the density function $f_a^s(X_a^s)$ at a current time. The above construction is done every δT time or in other words the time window W slides by δT time units.

We observe here that the above construction is based on the size of the window W. Choosing the right size of the time window W is extremely important for accurate construction of the *pdf*. If W is too large then the constructed *pdf* at time t may be significantly different from the true *pdf* at time t. For example, in a highway the arrival *pdf* at day time busy hours will be much different from that at middle of the night. Keeping the window length of about a whole day will not capture the true *pdf* at a particular time of the day. On the other hand if we have the time window very small, we may not have enough samples to construct the right *pdf*. Since the arrival of users is a stochastic process meaning that the arrival time *pdf* has time dependence, therefore the window size must be less than the period for which the process stays stationary at a given time. Based on these considerations, we choose a small window and use hysteresis or weighted informa-

[2]Slot index is always relative to the call set-up time, i.e., δt (not t) gives the slot index

tion about past *pdf* to construct the estimated *pdf* at current time. Let $f_a^s(X_a^s = k, t - \delta t)$ represent the *pdf* obtained at time $t - \delta t$ and in the current window R_k denotes the relative frequency of the event $(X_a^s = k)$, then estimated *pdf* at current time t is obtained as

$$f_a^s(X_a^s = k, t) = \alpha R_k + (1 - \alpha) f_a^s(X_a^s = k, t - \delta t)$$

where $\alpha \in [0, 1]$. The value of α near to 1 can be used if there are significant number of events taking place in the time window which may be true in rush hours. Otherwise, keeping α closer to 0.5 is suitable where the number of events occurring is less. The other strategy is to change the size of the window dynamically with changing traffic condition but we believe that in practice changing window will be more difficult than changing the value of α. Our experiments have shown that a given fractional change in window length will lead to more change in the measured value of the *pdf* than with the corresponding change in α.

In construction of the residence time *pdf* we only consider the departure events pertaining to a user handoff to another cell. Premature dropping or call termination events are not considered since such events do not reflect the locality based mobility behavior of users. Based on time occurrence of the departure events, the *pdf* for the residence time is constructed similar to the arrival time *pdf*.

V. Time based Reservation Framework

In this section we propose a time-based reservation framework where bandwidth is reserved on slot basis on the time domain using the optimal allocation algorithm discussed in section II. The framework is based on the advanced time reservation framework in fixed network as proposed in [14], [13]. First we describe the messaging and states required in the reservation setup. Next we discuss how the bandwidth becomes available to a handoff user based on this reservation framework.

A. Reservation Setup

User x currently located at cell s initiates the reservation by sending a reservation request to cell i where he wants reserve bandwidth. The reservation request denoted by RQST is a triplet $< s, B, d >$ where s is the origin cell where request is initiated, B is the bandwidth requested and d is the optional call residence time. The base station *bs* in any given cell keeps the following states to aid the reservation scheme: 1) a bandwidth state vector V of length l_{max} (refer to sec II) and 2) a set of slots for bandwidth allocation denoted by L_s per origin cell s as defined in sec II. In reference to the absolute time line of slots, the state vector V captures the bandwidth state starting from current time t (slot 0) to the future time t' (slot l_{max}). Therefore the vector V is updated at the end of every slot as $V[j] = V[j+1]\ \forall j = 0 \ldots (l_{max} - 1)$. In this way, $V[i]$ keeps the amount of bandwidth reserved in the i^{th} slot relative to slot 0, which corresponds to the current time.

Now, upon receiving RQST message $< s, B, d >$ from user x, bs at cell i finds out if there are available bandwidth to satisfy the users request in the slots given by L_s. This is done by checking the following condition given by

$$V[l] + B \leq C_i \ \ \forall l \in L_s$$

where C_i is the capacity of cell i. If the above condition is not satisfied, the *bs* sends a DENY message to the origin cell s for user x.

If the user x recives a DENY message from any cell, the call gets rejected. Otherwise, the user x sends a subsequent RESV message with the same information at RQST message to same cells where RQST message was sent. Receiving a RESV message, the *bs* reserves bandwidth by updating the vector V given as

$$V[l] = V[l] + B \ \ \forall l \in L_s.$$

B. Availability of bandwidth

When a user handoffs to a cell i, as a part of authentication procedure, the user passes the following information to the *bs*: 1) the origin cell id, 2) call originating time t_i. Based on the current time t and t_i, the *bs* finds out the slot offset l_{offset} by mapping the difference $t - t_i$ to integer slots. Finally from L_s, the *bs* finds out the slots $l' \in L'_s$ where $L'_s = \{l - l_{offset} \ | l \in L_s\}$. Therefore if the user stays in slots $l' \in L'_s$ and $l' \geq 0$, bandwidth gets available to the user by virtue of the above reservation scheme. On the other hand, if the user stays in slots where bandwidth is not reserved for him, the *bs* can either make bandwidth available from the unreserved pool of bandwidth, if available or terminate the connection.

C. An Example

Consider a slotted time domain with slot duration of unit time. Let the beginning of each slot be $t + i$ where i is an integer as shown in fig. 2. Consider users *U1*,*U2* and *U3* be located in cell 1,2 and 3 respectively. All the above users are suppose to request for reservation in cell i. We show how the bandwidth state vector V in cell i is updated with time. In fig. 2, V_t refers to the content/state of vector V at time t. Initially, at time $t + 1$, no bandwidth is reserved in cell i. At time *t1* user *U1* request for 3 units of bandwidth. Let $L_1 = \{1, 2, 4\}$ and therefore bandwidth is reserved in slot 1,2 and 4 as shown in updated V_{t1} (fig. 2). V_{t+2} and V_{t+3} shows how V gets updated at the beginning

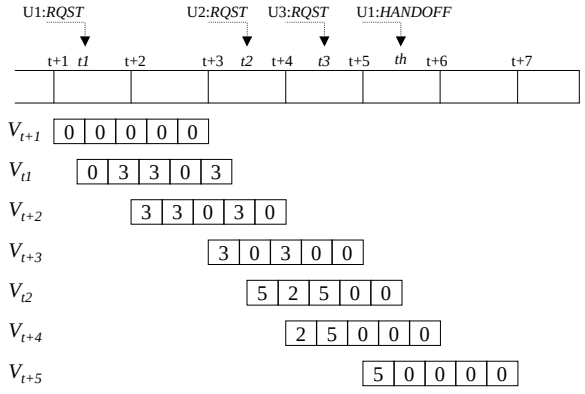

Fig. 2. Updating of V over time

of each slot. At *t2, U2* requests for 2 units and V is updated (as V_{t2}) by adding 2 units to respective slots given by $L_2 = \{0, 1, 2\}$. At *t3, U3* request for 2 units and let $L_3 = \{0, 3, 4\}$. Since capacity constraint is not met in slot 0, *U3* is rejected.

Now consider the user *U1* who handoffs to cell i at time t_h as shown in fig. 2. Mapping of $(t_h - t1)$ gives an offset of 4 slot from which we obtain $L_1' = -3, -2, 0$. The only valid slot is 0 where we can see from V_{t+5} that there is available bandwidth to support *U1*.

D. Implementation Issues

First we observe that the state space maintained by a single base station *b* is $(S + 1)l_{max}$ where S is the total number of possible call originating cell from where user may be expected to visit a given cell served by the *b.* Thus our proposed scheme is scalable as it doesn not require any per-user state. Secondly, the scheme does not employ any significant messaging or remote cell query except at the time of reservation set-up. It is also important to discuss the frequency at which the base stationof computes the allocation vector L_s for a origin cell s. In most real life cases the traffic pattern changes slowly and it is not necessary to compute L^s continuously. Therefore computation of L^s can be triggered when there is sufficient change in the arrival/residence time *pdf* expressed in terms of mean square error(*MSE*). *MSE* for two *pdfs* $f_i[1 \ldots l_{max}]$ and $f_j[1 \ldots l_{max}]$ is given as $\sum_{k=1}^{l_{max}} (f_i[k] - f_j[k])^2 / l_{max}$. Every δT interval of time when a new *pdf* is constructed, *MSE* is calculated with the last *pdf* used for computing L^s. If the $MSE \geq threshold$, a new L^s is computed where the value of the threshold is based on how fast is the reaction to changing traffic is wanted.

VI. Performance Evaluation Experiments

The main goal behind the experiments is to perform an extensive study about the properties of the proposed resource allocation scheme. Our major interest is directed at understanding how resource utilization depends upon the QoS level and the arrival/residence time density functions. As a part of evaluating each part of the scheme, we tried to focus on the performance of our density function measurement scheme and also on how our resource allocation scheme meets the target of providing QoS level to each individual user. Before going into discussing the experiments we first define the main performance metrics we are looking for.

- *% Bandwidth Used(%BU)* is defined as $\%BU = \sum_{i=1}^{N} B[i]/N$ where B is the reservation vector and N refers to the length of the reservation vector. *%BU* refers to the overall resource utilization of a cell.
- *Drop Ratio* is defined as the ratio of the number of hand-off users dropped due to unavailability of resources to the total hand-off users. The drop ratio does not refer to the *drop probability* which is an input to the resource allocation algorithm.
- *Blocking Ratio* is defined as the ratio of the number of new calls blocked to the total number of newly arrived calls.

A. Exact Arrival/Residence PDF

Here we are trying to evaluate the performance of the optimal algorithm in the ideal case where the exact arrival and residence time density functions are known for a given source cell. In order to obtain the required results we input the density function of arrival and residence time along with the target QoS level or *drop probability* to the allocation algorithm.

Based on the output of the algorithm, the %BU is obtained for different drop probability. The arrival and residence time density functions used in this experiment are shown in fig. 3. Figure 4 and 5 show the utilization versus the drop probability for residence time *pdf* I and II respectively. We observe that for a given drop probability, the utilization depends upon both the arrival and residence time *pdf*.

For example, we observe from fig. [4,5] that using uniform *pdf* for both arrival and residence time gives higher utilization than the exponential pdfs. We also note from comparing the results for arrival time *pdf* I and II that although both are having the same mean arrival time yet gives different utilization for a given QoS level. Most importantly, it is observed from the same comparison that more variance in the arrival time *pdf* generates lower uti-

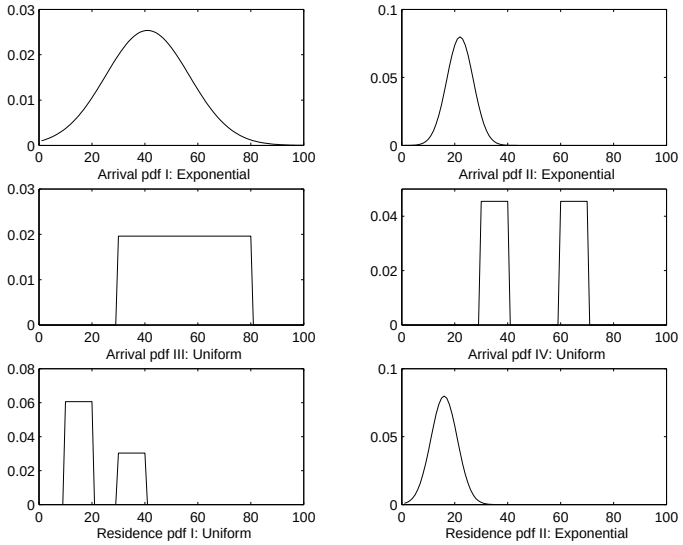

Fig. 3. Input exact arrival/residence time *pdf*

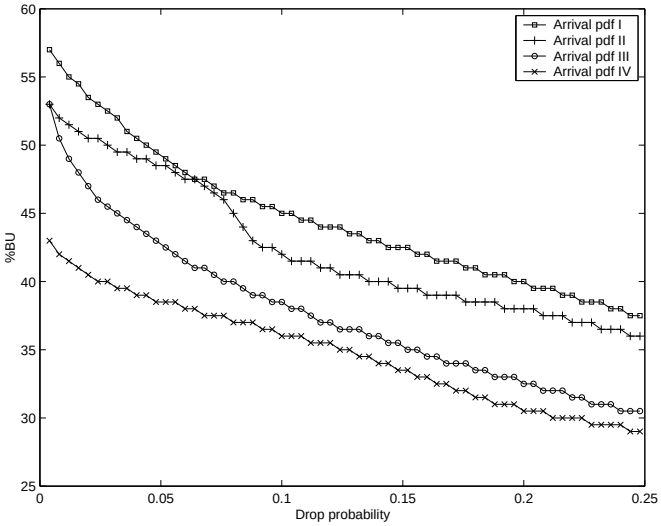

Fig. 5. % BU vs P_{drop} for residence time *pdf* II

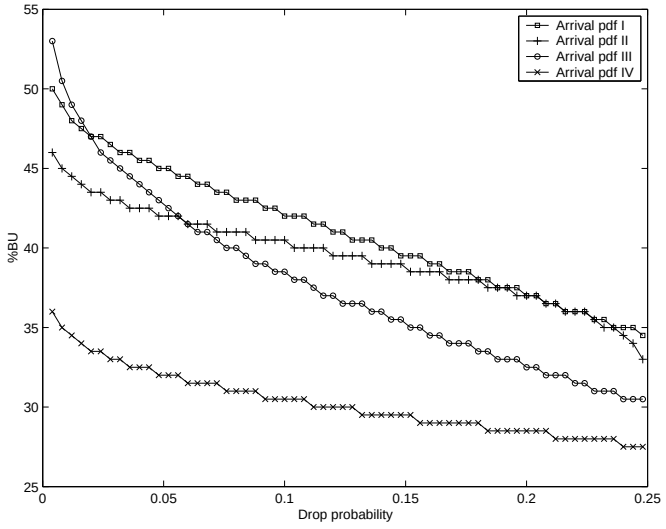

Fig. 4. % BU vs P_{drop} for residence time *pdf* I

lization. Applying the observation to real life scenarios suggests that in areas like highways where there is less variance in velocity, high utilization will be achieved with respect to crowded areas (near downtowns) with high variance in velocity. Although variance is a good measure to indicate the utilization level but it does not extend to the cases of bimodal density functions(arrival time *pdf* III). Although the variance of arrival time *pdf* III(2.0325e-04) is much higher than variance of *pdf* II(7.3422e-05), but utilization for arrival time *pdf* III is lower compared to that of arrival time *pdf* II. Arrival time *pdf* III being bimodal may represent a real life scenario of two possible independent behaviours of mobile users. But since the density function corresponding to individual bahaviours cannot be estimated and therefore it is difficult to characterize the effect of multimodal *pdf* on utilization.

B. Simulation Experiments

We conduct simulation experiments to explore more realistic scenarios on an event driven simulator. The specification of the parameters used in the simulator is given as follows. The time is quantized into slots of length T_L.. The time window of measurement(W) and value of α in *pdf* measurement are 500 secs and 0.8 respectively. The length of the reservation vector(l_{max}) is of length 250 slots. The call holding time is exponential distributed with mean 100 slots. Arrival of new calls follows a poisson process with mean rate λ.

Measurement of Density Functions:

The overall performance of the scheme and its optimality strongly depends upon how accurate the measured density functions are. Therefore we try to establish how far our measurement scheme meets the actual or true arrival time density function. In order to do that we consider two different scenarios where users from cell C1 are arriving at cell CM where the measurement is done. In the first scenario(SCEN1), the velocity of the users are considered exponentially distributed with a mean value of 40mph. In the second scenario(SCEN2), the velocity of users are assumed uniformly distributed with probability 0.7 from 40 to 60 mph and with probability 0.3 from 0 to 10 mph. We assume that the distance between C1 and Cm is 1 mile and users maintain a constant velocity during their call duration.

In cell CM we use our *pdf* measurement scheme based on the above assumptions and compare the measured *pdf* with actual *pdf* constructed based on data collected over the entire simulation time. For a simulation time of 20

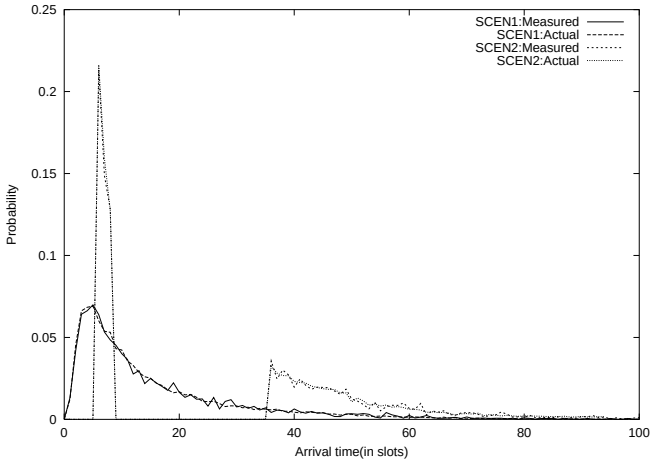

Fig. 6. Arrival time density function with $T_l = 10secs$

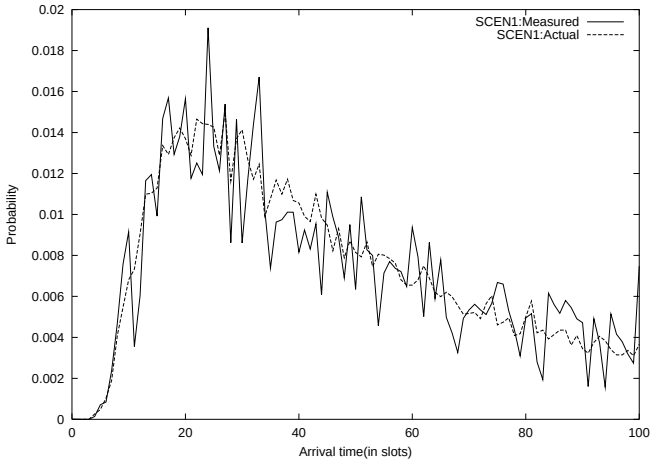

Fig. 7. Arrival time density function with $T_l = 2secs$

hours, we show the results in Fig. 6. The measured *pdf* in figure 9 refers to the *pdf* as measured at the end of 10 hours of simulation time. We observe that the measured *pdf* is not much different from the actual *pdf* for user arrival under both scenarios for $T_L = 10sec.$ From figure 7, we also observe that decreasing the slot length T_l introduces more spikes although retaining the same trend as the actual *pdf*. Such spikes represent inaccuracy in the measurement. That brings us to the conclusion that although decreasing slot size is favorable in giving higher utilization, it introduces measurement inaccuracy(for a given number of samples per window) and also higher time complexity for allocation algorithm and state space. Therefore slot length is an important trade-off in our proposed scheme and needs to be determined efficiently. There are statistical estimation techniques, for example the *Kernel Density Estimates*, which can be used to smooth the density function irrespective of slot size. However, this has not been considered in our present simulation experiments.

Measured Drop Ratio:

Our objective here is to find out how much difference exist between the measured drop ratio and the target drop probability. In the simulation model, we consider the scenario SCEN2 again and assume that the cell CM has infinite capacity and λ to be 0.7. Since the drop rate is not dependent upon the capacity of the cell but on the reservation vector and therefore infinite capacity assumption is valid. We also assume that T_L = 10sec. Our measurement of drop ratio is done over the entire simulation time of 20 hours and for each simulation run, the target drop probability is varied. From fig. 8, we observe that the difference between the measured drop rate and the target drop probability is not significant. Therefore, small inaccuracies as shown in fig. 6 related to the measurement do not affect significantly the achieved drop rate. The little difference between the measured and the target drop rate can be caused by the threshold used for triggering the reservation vector computation. Choosing a smaller value of the threshold K may result in lower difference but frequent computation of reservation vector will be necessitated.

Utilization Comparison:

Our intention is to find out the dependence of the utilization on the achieved drop ratio which takes into consideration the inaccuracies involved in the measurement of the *pdfs*. We compare the utilization based on using the measured density function with the utilization based on using the actual density functions. The simulation model for this experiment considers scenario SCEN2 and also assumes infinite cell capacity for reason stated above. For each simulation run we vary the target drop probability to obtain the graph as shown in fig. 9. From the graph we observe that the measured density functions results in a lower utilization compared to the actual density functions. But the observed difference is less than 5 percent and does not vary with the target QoS level.

New Call Blocking Ratio:

New call blocking ratio depends upon the way the reservation is made on time. The simulation scenario considered here consist of 8 cells C1 to C8 from where users try to reserve bandwidth at cell CM. Velocity of users follows from scenario SCEN2 and each cell is at a different distance from CM. The bandwidth requirement of the mobile can be 1,2,4 and 8 bandwidth units with probability 0.5,0.3,0.1 and 0.1 respectively. We also assume that the cell capacity is 40 bandwidth units and the target drop probability to be 0.1 for the entire simulation run. Based on the above simulation model, we intend to compare the new call blocking ratio of our scheme to that of a non time-based scheme where resources are reserved for the entire duration of the call.

For each simulation run of 20 hours we vary the value

of the offered load λ to obtain the variation of new call blocking ratio with the offered load as shown in Fig 10. We observe that our scheme achieves much lower new call blocking rate than the non time-based scheme. An important point to note here is that the new call blocking rate achieved in this scheme does not completely depend upon the resource utilization. New call may be dropped due to bandwidth fragmentation in time as a consequence of our temporal reservation scheme. But high resource utilization may lead to an increase in the admitted traffic if we consider the traffic to be composed of different classes of users which includes users not in need of strong bandwidth guarantees.

Time Varying Mobility:

In this experiment, we consider the two cells C1 and CM again with the assumption that the arrival time distribution of the users is uniformly distributed over a period [t1:t2] secs. Both t1 and t2 varies with time in the following way in the simulation experiments. For every 10000 secs simulation time, t1 is also varied randomly(with uniform distribution) selected from 0 to 1000secs and t2 is randomly(with uniform distribution) from v1 to 1000secs. In such a time varying mobility scenario, our objective is to find out how the measured drop rate varies with time. We define the cumulative drop rate at a given time t as the ratio of the total number of handoffs drops to total number of handoffs in the interval [0,t]. The instantaneous drop rate at a given time t is defined as the total number of handoff drops to total handoff in the interval [t-w:t] where w is a constant time window of 500 secs. In the simulation experiment we have kept target drop probability of 0.1. Figure 11 shows the temporal behavior of the cumulative and the instantaneous drop rates. We observe that the cumulative drop rate slowly approaches towards the target drop rate with time. This implies that over a sufficient amount of time, the scheme achieves almost the target drop rate. We also observe that there is large variation in the instantaneous drop rate with time. The spikes in the instantaneous drop rate curves indicate the time when there was change in the mobility scenario. The spikes are caused when an allocation vector corresponding to an old arrival density function is used for the current users. The sharpness of the spikes indicates that the scheme reacts fast to changing the mobility scenario.

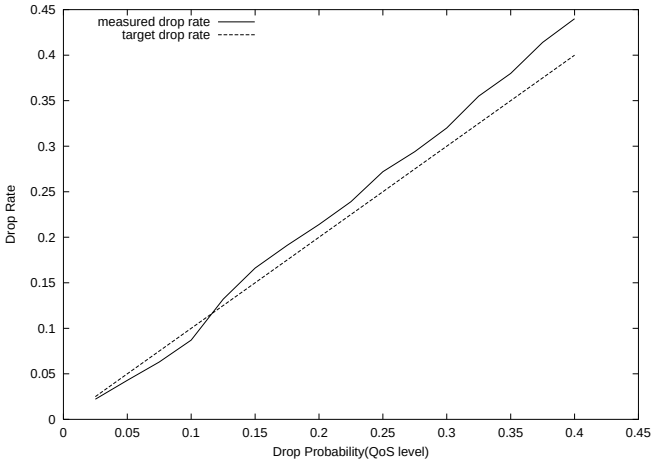

Fig. 8. Drop rate vs target QoS level

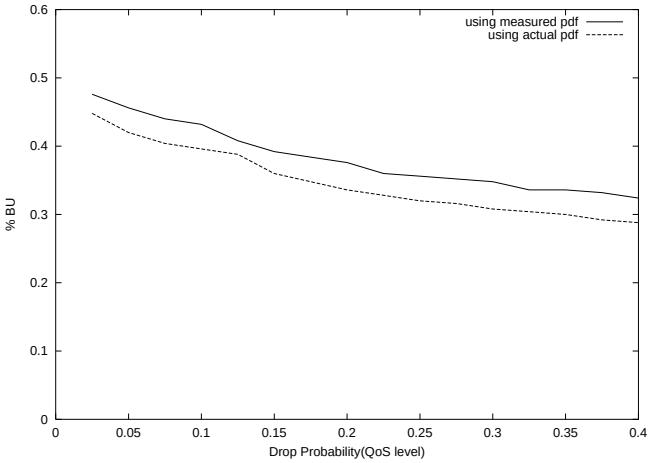

Fig. 9. Utilization vs drop probability

VII. Conclusion and Future Work

The objective of the work presented in the paper is to explore the time-based resource allocation problem to increase the utilization of a cellular network. Our work in this regard resulted in the following main contributions: (1) an algorithm for finding the optimal bandwidth allocation in time. (2) a measurement scheme to construct arrival/residence time distribution based on just monitoring the handoff events and (3) a time-based resource reservation framework.

Based on simulation results, we have shown that optimal utilization of a single cell depends strongly on both the target QoS level(drop probability) and the arrival/residence time distribution. The results confirm the fact that a scheme which does not incorporate the arrival/residence time distribution and the QoS level is not likely to result in efficient utilization. Further, the present studies also reveal that despite the little inaccuracies in our measurement process, the proposed scheme still achieve the target QoS level with near to optimal utilization.

Our further extension of our work will involve taking more realistic scenarios as mentioned in [12] and use different spatial resource allocation schemes along with our scheme to find how they work together under varied mobility patterns. Also this work focus only on a class of applications needing hard QoS guarantee. A more inter-

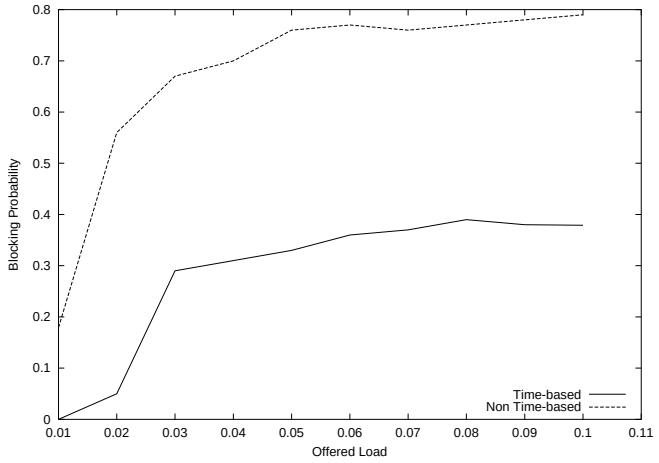

Fig. 10. Call blocking rate vs offered load

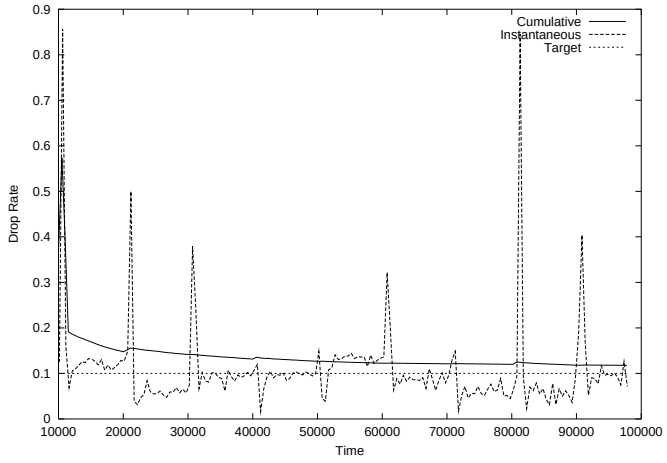

Fig. 11. Drop ratio with time

estig work will address how to allocate resource based on time windows for bandwidth adaptive applications.

REFERENCES

[1] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "On accomodating mobile hosts in an itegrated service packet network," in *Proc. IEEE INFOCOM'97*, pp. 1048-1055, April 1997.

[2] S. Lu, V. Bharaghavan, and R. Srikant, " Adaptive resource management for indoor mobile computing environments," in *Proc. of ACM SIGCOMM '97*, France, September 1997.

[3] A. Aljadhai and T. Znati, "A Framework for Call Admission Control and QoS Support in Wireless Environments", in *IEEE INFOCOM'99*, New York, March 1999.

[4] Y. Zhao, *Vehicle Location and Navigation Systems*, Artech House, 1997.

[5] M. Naghshineh and M. Schwartz, "Distributed Call Admission in Mobile/Wireless Networks," *IEEE Journal for Selected Areas in Communications*," 14(4), pp. 711-717, 1996.

[6] A. Acampora and M. Naghshineh, "Design and control of microcellular networks with QOS provisioning for data traffic," *Wireless Network*, vol. 3, pp. 249-256, September 1997.

[7] R. Ramjee, R. Nagarajan, and D. Towsley, " On optimal call admission in cellular networks," in *Proc. IEEE INFOCOM '96*, pp. 43-50, San Francisco, March 1996.

[8] C. Chao and W. Chen, " Connection admission control for mobile multiple-class personal communication networks," *IEEE Journal on Selected Areas Communications*," 15(8), pp. 1618-1626, 1997.

[9] S. Ganguly, D. Niculescu and B. Vickers, "Dynamic QoS Provisioning in Wireless Data Networks," in *Proc. IEEE VTC 2001*, Greece, 2001.

[10] D. Levine, I. Akyildiz, and M. Naghshineh, " A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Trans. on Networking*, vol. 5, pp. 1-12, February 1997.

[11] S Choi and K. G. Shin, "Predictive and Adaptive Bandwidth Reservation for Hand-Offs in QoS-Sensitive Cellular Networks," in *Proc. ACM SIGCOMM'98*, pp. 155-166, Vancouver, September 1998

[12] R. Jain and E.W. Knightly, "A Framework for Design and Evaluation of Admission Control Algorithms in Multi-Service Mobile Networks," in *Proc. IEEE INFOCOM '99*,New York, March 1999

[13] R. Guerin and A. Orda, "Networks With Advance Reservations: The Routing Perspective," in *Proc. INFOCOM'00*, Israel, March 2000.

[14] M. Degermark, T. Khler, S. Pink and O. Scheln, "Advance Reservations for Predictive Service," *NOSSDAV 1995*, pp. 3-15.