Optimal Bandwidth Reservation Schedule in Cellular Networks

Samrat Ganguly   Badri Nath   Navin Goyal
Department of Computer Science
Rutgers University, Piscataway, NJ, 08852

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 finding the reservation schedue that would minimize 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.**

Index Terms— **Cellular Networks, Mobility, Reservation, Optimization**

I. Introduction

A. Background

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 requirements of the above applications the service providers ought to adopt some form of a reservation scheme or a 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 made available to support the mobile user else the user will suffer a forced termination of his call in progress. Therefore, careful 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 of any resource allocation scheme. Achieving the above goal provides the probablistic quality of service (QoS) guarantee as desired by a mobile user. However, 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], [3], direction prediction [4], knowledge about possible geographic routes with the help of ITS Navigation system [5], [6] etc. The objective of most of these schemes is to select the cells where bandwidth reservation need to be made.

Temporal resources on the other hand refers to the amount of time the resources are reserved in these selected cells and expressed in terms of the *time-bandwidth* product. For example, if a connection reserves B bandwidth units for t units of time in cell s, then $B * t$ amount of resources gets used on behalf of the connection in cell s. Clearly, minimizing the *time-bandwidth* product per user in each cell should also be an objective of any reservation scheme. However, to maintain the QoS, minimization of the *time-bandwidth* product must meet the drop probability requirement of a connection.

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 a challenging task to minimize the time for which bandwidth reservation must be held in the cells for the user. To this end, 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.

This research work was supported in part by DARPA under Contract number N-666001-00-1-8953

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

B. Related work and motivation

Majority of the earlier research in the area of resource allocation was based purely on call admission control without keeping any reservation states. These schemes such as in [7], [8], [9], [10], [11] were mostly based on either dynamical or statical prediction of the steady state distribution of users' demand in different cells. In contrast, in the recent past, several schemes based on keeping reservation states and per user monitoring were proposed in [1], [4], [3], [12] and found to perform better than the above schemes based on simulation experiments presented in [13]. Some of these reservation based scheme such as in [1] were just based on estimating spatial per user resource demand while others as in [4], [3], [12] 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. Futhermore, in explicit reservation based approaches where bandwidth is simply reserved in cells, the problem of time management of the bandwidth resources has not received attention in the existing literature. Time management of resources leads us to a range of following questions: How long do we reserve resources in any given cell? Can we minimize the length of the reservation in time by using any mobility related information? Can the input QoS parameter such as drop probability be realized using reservation in time domain? How does users mobility behaviours affects the resource allocation in time? Focus of this work is trying to understand and answer these questions.

C. Contributions

The main contribution of our work is to develop an optimal scheme for resource allocation that finds a reservation schedule which minimizes the amount of time resources gets 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 probability of arrival, the arrival and the 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 to 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. We further provide an extensive set of simulation results that lets us understand how the different mobility behaviour affect the bandwidth reservation in time and show the performance of the scheme under inaccurate arrival/residence probability distribution of users.

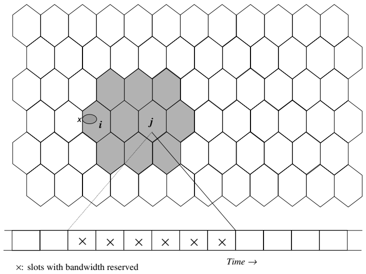


Fig. 1. Bandwidth reservation at cell j in space and time

D. Organization of the paper

The rest of the paper is organized as follows. In the next section, we discuss the issues of resource allocation particularly focussing on allocation in time domain and finally formulating the optimization problem. In section 3, we present the algorithm for obtaining the solution to the optimization problem. We present a scheme for constructing the arrival/residence time distribution in section 4. We present our proposed time-based reservation framework in section 5. Simulation results are presented in section 6. Finally the the main conclusion is drawn in section 7.

II. Resource Allocation in Space and Time

In this section, first we describe system model of a cellular system with bandwidth reservation. We then discuss in general sense the resource allocation problem finally focussing on the problem of allocation in time domain.

In a cellular network, let cell i be the current location of the mobile user x as shown in figure [1]. Let C be the set of cells where mobile user x requests to reserve resources $D(x)$ as shown by shaded region in figure [1]. $D(x)$ refers to the effective bandwidth [14], [15], [16] requirement of the user computed based on users individual requirement regarding channel quality, delay requirements etc. The set of cells C correspond to the spatial component of the resources that maybe reserved on behalf of the user. Minimizing the number of cells in C will therefore constitute an objective towards increasing overall utilization. Selection of these cells for reservation must consider predicting users' mobility profile, direction, velocity and call duration. Reservation of bandwidth in multiple cells at call set-up is discussed further in [17]. Once the cells in C have been identified, the next important step would be to find how to reserve the bandwidth over time in each of these cells in C. 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. Consider a

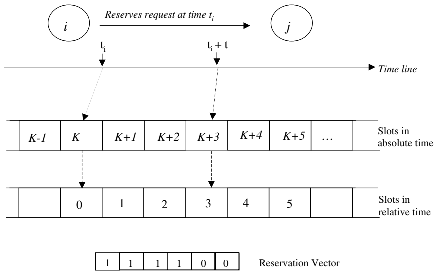


Fig. 2. Reservation in slots in the relative time frame

single cell j in C, a particular case of reservation over time is shown in figure [1]. If n_j refers to the total number of slots where resources needs to be reserved in cell $j \in C$, then total bandwidth resources reserved for the given user x is given by $R(x) = \sum_{j \in C} D(x) \times n_j$. Thus, minimizing $R(x)$ should be the goal of any resource reservation scheme.

A. Allocation problem in time domain

The allocation problem in time domain relates to finding the exact *reservation schedule* for a given user in each cell $j \in C$. By *reservation schedule,* we refer to the time slots where bandwidth needs to reserved on behalf of the user. A *reservation schedule* for a given user x is derived from a *Reservation vector B* which is defined as follows.

Definition : A reservation vector B_j^x is a binary vector of length N where the i^{th} position refers to the i^{th} time slot in relative time frame where slot 0 refers to the current time t_0. $B_j^x[i] = 1$ implies that resource is reserved in the i^{th} time slot and $B_j^x[i] = 0$ implies otherwise.

The concept of relative time frame is shown in figure [2]. As shown, a reservation request made by user x currently at cell i to cell j at time t_i is mapped to the slot index 0 in the relative time frame. In general, if t_{max} be the maximum call duration of a user, in that case, $t_i + t_{max}$ is mapped to slot index N[1]. N refers to the number of slots corresponding the maximum call duration. Therefore, the reservation vector B_j^x denotes the the time window of reservation for user x in cell j. A *reservation schedule L* for a user x refers to the set of slots where $L = \{i \,|B_j^x[i] = 1\}$. Figure [2] shows a request that arrives at current time t_i for reservation from t_i to $t_i + t$ and the corresponding reservation vector B_j^x. The figure also shows the slots in the real time domain in cell j where the resources must reserved.

Our objective here is to find the *reservation schedule* for a given cell j that would minimize the number of slots n_j and meet the drop probability requirement of the user x. We therefore show two scenarios where the *reservation schedule* maybe computed.

Ideal Case: In the ideal situation, the *exact* mobility profile of the user x may be known apriori at call set-up time.

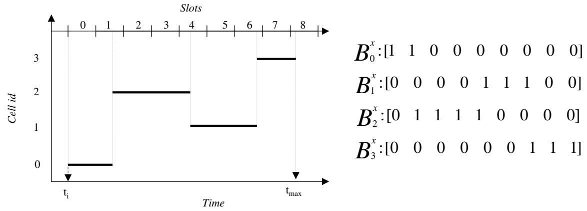

Slots
0
1
2
3
4
5
6
7
8
Cell id
3
2
1
0
t_i
t_{max}
Time
B_0^x:[1 1 0 0 0 0 0 0 0]
B_1^x:[0 0 0 0 1 1 1 0 0]
B_2^x:[0 1 1 1 1 0 0 0 0]
B_3^x:[0 0 0 0 0 0 1 1 1]

Fig. 3. Ideal scenario of bandwidth reservation in time

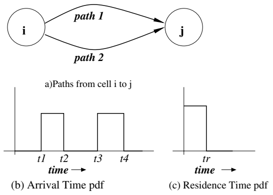
An *exact* mobility profile would consist of the cells that the user is going to visit along with the exact time of arrival and departure in each of these cells (fig. [3]). In such a case finding the reservation vector is trivial and is shown in figure 3 for each cell.

Real-life Case: In the practical situation, such an *exact* mobility profile for a given user may not be assumed to be known at call set-up time. However, a more realistic mobility profile of a user that may be known apriori can be characterized in the following probabilistic terms.

- **Probability of Arrival ($p_{x,j}$)**: In a realistic scenario, user x may not visit all the cells in C in his call duration. Instead, the user will have a probability of arriving at a cell $j \in C$ denoted by $p_{x,j}$.
- **Arrival time probability density function ($f_a^{x,j}(.)$)**: Although the exact time of arrival for user x at cell j may not be known apriori, but one can assume that the arrival time of the user is likely to follow a stationary distribution. In other words, the user may arrive in different time slots with different probabilities. To express the above characterization, we 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 $f_a^{x,j}(X_a)$ as the corresponding arrival time probability density function(*pdf*) of user x in cell j. $f_a(X_a)$ is a discrete function with the property that $\sum_0^N f_a^{x,j}(X_a) = 1$.
- **Residence time probability density function ($f_r^{x,j}(.)$)** Similar to the arrival time, the residence time of a user x can expressed in terms of the probability density function. We therefore define the random variable $\mathbf{X}_r = n$ as outcome that user x departs the cell j at the n^{th} time slot. $f_r^{x,j}(X_r)$ thus denotes the corresponding discrete residence time *pdf* for $\mathbf{X}_r$.

The probability of arrival along with the arrival/residence *pdf* constitutes the probabilistic mobility profile (PMF) of a user x. At this point, we assume that such a PMF for an user is known to us and discuss importance and usability of it in finding the reservation schedule. We defer the construction of the PMF to section 4. For the time being let us assume that $p_{x,j} = 1$ and discuss the issues in finding the reservation schedule based on the arrival/residence *pdf* through the following example.

[1] There is no reason to keep a reservation state beyond the call duration time in any cell



a)Paths from cell i to j

(b) Arrival Time pdf

(c) Residence Time pdf

Fig. 4. Example of a uniformly distributed Arrival/residence time

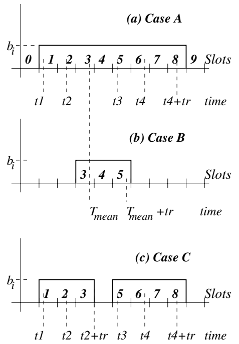


Fig. 5. Slots for reservation schedule

B. An Example

Figure 4(a) shows a user x currently at cell i and can arrive at cell j in two different paths. The arrival time at cell j along each path is uniformly distributed and the resulting the arrival time pdf shown in figure 4(b). Similarly the residence time is also uniformly distributed as shown in figure 4(c). Based on the given *pdfs,* we consider the following cases of finding reservation schedule.

- Case A: Since the earliest and latest arrival time for fig. 4(b) is $t1$ and $t4$ respectively and also the latest departure time is tr, resources can be reserved simply from slot corresponding to $t1$ to slot corresponding to $t4+tr$. as shown in figure 5(a). In that case user will be guaranteed availability of bandwidth and we note that $n_j = 8$ [fig. 5(a)].
- Case B: Another possibility maybe to reserve bandwidth from the mean arrival time till latest departure time as shown in figure 5(b). In that case although n_j is greatly reduced but about half of the users coming between time [t1:t2] will be dropped. Obviously this is not a practical possibility.
- Case C: Here we take a closer look at arrival pdf and based on its nature find the allocation slots. From figure 5(c) we see that there is no point in allocating in the slot 4 where there is no chance for a user to stay.

Although in case C, we were able to identify slots where probability of a user staying is zero but such case may not exist for most arrival/residence time *pdfs.* In a general case, for every slot there might some nonzero probability for a user to stay. Under such a case how do we find the right slots to reserve? We cannot simply use a strategy of excluding slots for allocation with zero probability of users staying. Also in above cases A,C we tried to obtain a drop probability of zero. But what will be the reservation schedule if the required drop probability is not zero but some value greater than zero. Therefore it is difficult to find out the right reservation schedule that will assure a level of QoS(drop probability) along with minimizing the allocated slots. In the next section we provide a formal specification of the problem.

C. Problem Statement

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 n_j in a given cell j 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, we define a *projection vector P* for a given the reservation vector B_j^x.

Definition : A projection vector is a binary vector of lenght N denoted as $P[0 \dots N]$ where $P[i]$ is defined as follows.

$$P[i] = 0 \quad \textbf{IF} \quad B_j^x[i] = 0$$

$$P[i] = k \quad \textbf{IF} \quad \forall\ j = i \dots i + k - 1,\ B_j^x[j] = 1$$

Therefore, $P[i]$ basically denotes the number of consecutive *1's* starting from the i^{th} position in B_j^x. 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[2]. 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

$$P_{drop}(B_j^x) = p_{x,j} \times \sum_{i=0}^{N} f_a(i) \times P_{cdrop}(i). \qquad (1)$$

We observe that the drop probability depends upon B_j^x and the total resources allocated for a user n_j is given by $\sum_{i=0}^{N} B_j^x[i]$. For ease of presentation we omit the subscript/superscript of B_j^x henceforth. 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.

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

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

$$P_{drop}(B) < T_{QoS}$$

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

[2] 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

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. For the ease of presentation, we rewrite the above optimization problem by redefining P_{drop} as

$$P_{drop}(B) = \sum_{i=0}^{N} f_a(i) \times P_{cdrop}(i). \tag{2}$$

and the contraint equation as

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

where $T = T_{QoS}/p_{x,j}$. Initially we consider $B[i] = 1$ for all i, which results in $P_{drop}(B) = 0$ from (2). 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 the combinations of k zeros in B the 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

$$P_{drop}(B) = P_{drop}(B[0\ldots i-1]) + f_a(i) + P_{drop}(B[i+1\ldots N]) \tag{3}$$

Proof: From eqn(2), we can express $P_{drop}(B)$ as follows:

$$P_{drop}(B) = \underbrace{\sum_{j=0}^{i-1} f_a(j) \cdot P_{cdrop}(j)}_{X}$$

$$+ \underbrace{f_a(i) \cdot P_{cdrop}(i)}_{Y} + \underbrace{\sum_{j=i+1}^{N} f_a(j) \cdot P_{cdrop}(j)}_{Z}$$

Since $P[j]$ and hence $P_{cdrop}(j)$ in X does not depend upon the values in positions $i \ldots N$ of B because of the zero in the i^{th} position, therefore X 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, Z is equal to $P_{drop}(B[i+1\ldots N])$. Finally, since $P[i] = 0$ which implies $P_{cdrop}(i) = 1$ thus making Y being equal to $f_a(i)$, hence proves the proposition. ■

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 $A_0^k \ldots A_N^k$ where A_i^k is $B_k[i\ldots N]$. 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. The position p is obtained where by inserting a zero minimizes the value of $P_{drop}(A_i^1)$. We next update A_i^1 by inserting a zero in the p^{th} position. We also obtain $POS(A_i^1) = POS(A_i^0) \cup p$. Likewise we compute A_i^1 for $i = 0 \ldots N$. 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 probability is a minimum. Therefore, we assign $B_1 = A_0^1$. We move into the second iteration if $P_{drop}(B_1) < T$.

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 *two* zeros to be inserted in A_i^2 which minimizes $P_{drop}(A_i^2)$. In other words, there are $C(N-i+1, 2)$ possible combinations of inserting *two* zeros in A_i^2. We want to find out the particular combination for which $P_{drop}(A_i^2)$ is minimum. 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 (3). 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 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. It may be noted that the second zero must be in position $p + POS(A_{p+1}^1)$ which we already found in the 1^{st} iteration. For example, consider $N = 5$, then following are A_0^2 for different value of p (the first position of zero).

$$p = 0 \rightarrow A_0^2 = [0, A_1^1]$$

$$p = 1 \rightarrow A_0^2 = [1, 0, A_2^1]$$

$$p = 2 \rightarrow A_0^2 = [1, 1, 0, A_3^1]$$

$$p = 3 \rightarrow A_0^2 = [1, 1, 1, 0, A_4^1]$$

$$p = 4 \rightarrow A_0^2 = [1, 1, 1, 1, 0, A_5^1]$$

$$p = 5 \rightarrow A_0^2 = [1, 1, 1, 1, 1, 0]$$

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 (1), we can write $P_{drop}(B_{k+1})$ as

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

If $P_{drop}(B_{k+1}) < P_{drop}(B_k)$, it follows $c1 + c2 + c3 < P_{drop}(B_k)$ from (5). 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: $B = B_k$ achieves minimum of $P_{drop}(B)$ using k zeros where B_k is found by the above algorithm.

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 (3) plays the crucial role in the induction step. If $k = N$ then there is nothing to prove. So assume that $k < N$, in that case there is some index i with i^{th} entry 0 in B. In the algorithm, we try all the N 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. ■

Propositions (2) and (3) 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 where $P_{drop}(B_k) < T$.

A. Complexity

Proposition 4: The algorithm has a space complexity of $O(N^2)$ and a time complexity of $O(i*N^2)$ where i is the stopping iteration.

Proof sketch: 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 $O(i * N^2)$.

B. Modified Algorithm with Slot Restrictions

In the above algorithm we assumed that bandwidth is available in all slots in vector B for the request. Therefore, if in some slots in the computed *reservation schedule* bandwidth is not available, the request is blocked. But it is still possible to compute a admissible *reservation schedule* by modifying the algorithm to include the slot restrictions where bandwidth cannot be reserved. In the modified algorithm, we create a new vector B' of length N' where N' is the number of slots in the orginal vector B where bandwidth is available for the request. We also define another vector M of length N' which maps the slots in B to corresponding slots in B'. For example $M[i] = j$ means that i^{th} slot in B correspond to the j^{th} slot in B'. Our next step would be to compute B' such that $\sum_{i=1}^{N'} B'[i]$ is minimized subject to the drop probability constraint. In order to compute B' we need to compute the The drop probability for a given B' is computed by first obtaining B from B' and then using equation (1). B can be obtained from B' and M as $B[i] = 1$ if and only if $B'[M[i]] = 1$. For computing B' we follow exactly same steps as for computing B. It can be easily verified that above algorithm is correct since both proposition 1 and 2 hold for B' (formal details of the proof is same as for B and thus omitted).

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. Construction of the probabilistic mobility profile

The probability of arrival for a user $P_{x,j}$ can be computed based on the knowledge of the geographical location of the current cell and cell j, velocity of the user, existing geographical route and past monitored mobility profile. Schemes using the above knowledge used prediction to find out probability of a user to visit cells in the adjacent region. Details about such location prediction schemes can be found in [3], [4], [6].

A significant amount of research has also been done is computing the probabilistic models about the arrival time and residence time probability distribution [19], [20], [21]. The general approach tries to collect statistics of multiple users in the region and fit the information into known probabilistic model. The assumption in using a probability distribution to predict users movement is that such distribution follows a stationary stochastic process.

Since our algorithm works on any general arrival/residence time distribution, therefore it is not necessary that a probabilistic model needs to be used for computing the reservation schedule. Rather, distribution function constructed empirically can achieve much better approximation to the actual mobility profile. Next we discuss the issues and construction of such a distribution function.

A. Source cell based Arrival/Residence *pdf*

From the definition of the arrival time *pdf* in section 2, we see that it refers to the probability 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*.

B. 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 originating 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[3]. 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 $k = 0, \ldots, N$ 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 information 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

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

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 [22], [23]. 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 $< s, D(x), d >$ where s is the origin cell where request is initiated, $D(x)$ 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 N (refer to sec II) and 2) a *reservation schedule* 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 N). Therefore the vector V is updated at the end of every slot as $V[j] = V[j+1]\ \forall j = 0 \ldots (N-1)$. 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 $< s, D(x), d >$ 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

$$V[l] + D(x) \leq C_i \quad \forall l \in L_s$$

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 receives a DENY message from any cell, the call gets rejected. Otherwise, the user x sends a subsequent RESV message with the same information as the 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

$$V[l] = V[l] + D(x) \quad \forall l \in L_s.$$

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.

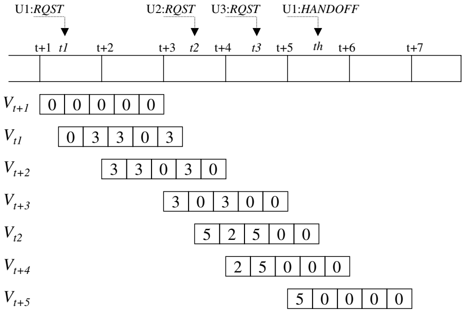


Fig. 6. Updating of V over time

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. 6. 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. 6, 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. 6). V_{t+2} and V_{t+3} shows how V gets updated at the beginning 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. 6. 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 $(S+1) \times N$ 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 does not 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 station 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* $f_i[1 \dots N]$ and $f_j[1 \dots N]$ is given as $\sum_{k=1}^{N}(f_i[k] - f_j[k])^2/N$. 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 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 utilized for a user.
- *% Utilization* is defined as $1 - n_j/n_a$ where n_a is the actual number of slots used by the user when he visited the cell j.
- *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.

The arrival and residence time density functions used in this experiment are shown in fig. 7. Figure 8 and 9 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 %BU depends upon both the arrival and residence time *pdf*.

For example, we observe from fig. [8,9] that using uniform *pdf* for both arrival and residence time gives higher utilization than the exponential pdfs. We also note from comparing the

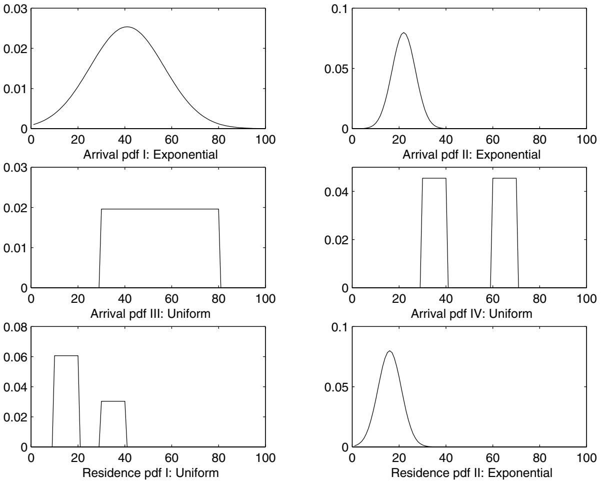


Fig. 7. Input exact arrival/residence time *pdf*

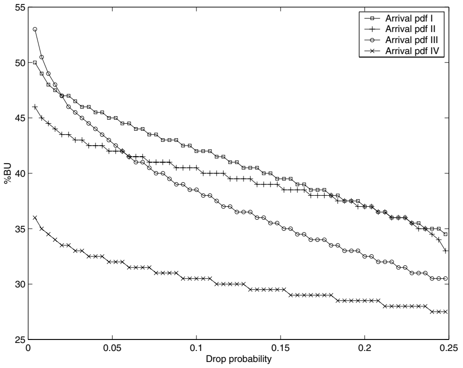


Fig. 8. % BU vs P_{drop} for residence time *pdf* I

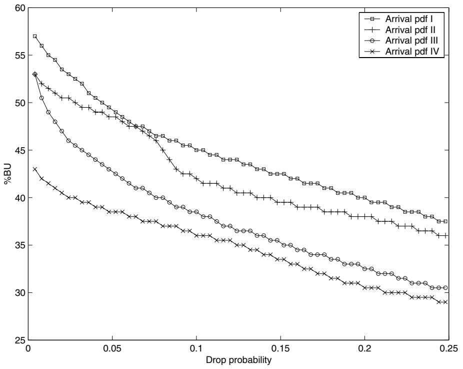


Fig. 9. % BU vs P_{drop} for residence time *pdf* II

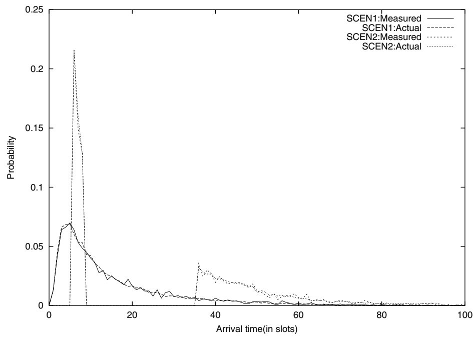


Fig. 10. Arrival time density function with $T_l = 10secs$

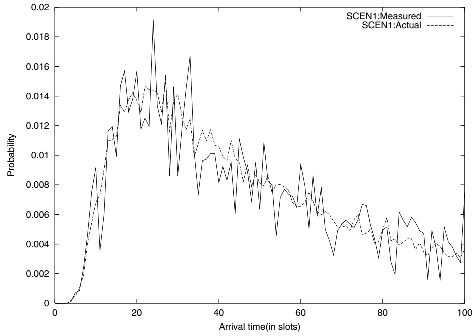


Fig. 11. Arrival time density function with $T_l = 2secs$

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 utilization. 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* IV). Although the variance of arrival time *pdf* II(2.0325e-04) is much higher than variance of *pdf* IV(7.3422e-05), but utilization for arrival time *pdf* II is lower compared to that of arrival time *pdf* IV. 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(N) 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.

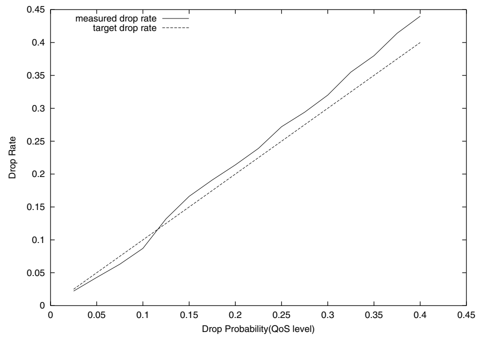


Fig. 12. Drop rate vs target QoS level

Measurement of Density Functions: The overall performance of the scheme 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 C_1 are arriving at cell C_M 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 C_1 and C_M is 1 mile and users maintain a constant velocity during their call duration.

In cell C_M 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 hours, we show the results in Fig. 10. The measured *pdf* in figure 10 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 11, 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 and 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.

Measured Drop Ratio: Our objective here is to find out how much difference exists 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 C_M has infinite capacity, λ to be 0.7, T_L = 10sec and simulation time of 20 hours. Since the drop rate is not dependent upon the capacity of the cell but on the reservation vector and therefore

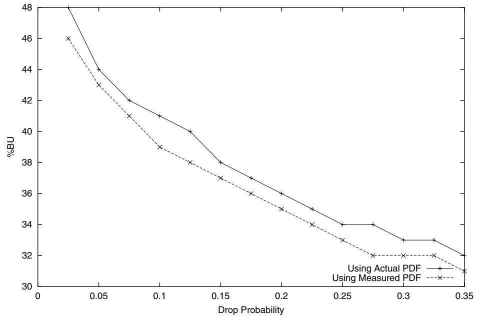


Fig. 13. %BU vs drop probability

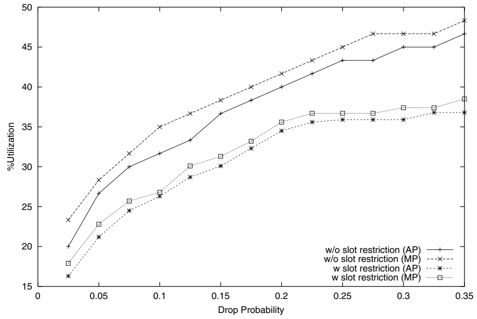


Fig. 14. %Utilization vs. drop probability

infinite capacity assumption is valid. From fig. 12, 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. 10 related to the measurement do not affect significantly the achieved drop rate.

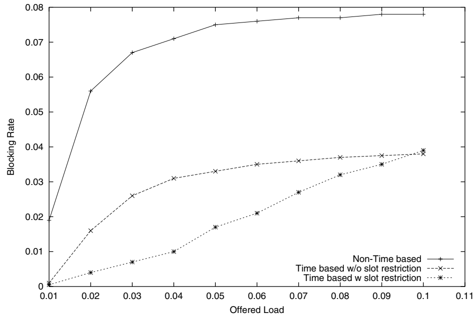


Fig. 15. Call blocking rate vs offered load

Bandwidth Utilization Comparison: Our intention is to find out the dependence of the %BU on the achieved drop ratio which takes into consideration the inaccuracies involved in the measurement of the *pdfs*. 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. 13. 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.

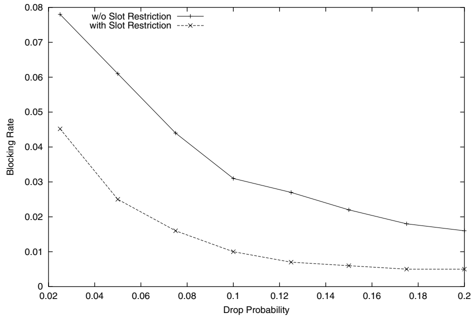


Fig. 16. Blocking rate vs. drop probability

Fig 14 also shows the %Utilization obtained from %BU compared for two version of the algorithm where one considers slot restriction and other does not. We show the %Utilization curve follows from the %BU curve with higher utilization in case of the algorithm without slot restriction.

New Call Blocking Ratio: The simulation scenario considered here consist of 8 cells C_1 to C_8 from where users try to reserve bandwidth at cell C_M. Velocity of users follows from scenario SCEN2 and each cell is at a different distance from C_M. 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 with target $P_{drop} = 0.1$. 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 offered load λ for the results as shown in Fig 15. 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 is not just a function of %BU but also depends upon the bandwidth fragmentation in time. Due to this effect, we observe that the algorithm with slot restriction has lower call blocking at lower load. At higher load, though the difference becomes less. A possible reason may be that at higher load there is more requests leading to higher probability in filling up the holes created due to bandwidth fragmentation. Figure 16 also shows that the call blocking rate in the case of the algorithm without slot restriction is higher than that with slot restriction for different drop probabilities.

Time Varying Mobility: In this experiment, we consider the two cells C_1 and C_M again with the assumption that the arrival time distribution of the users is uniformly distributed over a period $[t_1 : t_2]$ secs. Both t_1 and t_2 varies with time in the following way in the simulation experiments. For every 10000 secs simulation time, t_1 is also varied randomly(with uniform distribution) selected from 0 to 1000secs and t_2 is randomly(with uniform distribution) from v_1 to 1000secs. In such a time varying

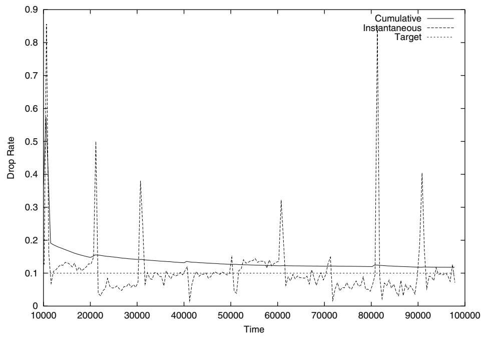


Fig. 17. Drop ratio with time

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 17 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. The spikes in the instantaneous drop rate curves indicate the time when there was change in the mobility scenario.

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 [13] 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 interesting work will address how to allocate resource based on time windows in a multiple class scenario with coexistence of bandwidth adaptive applications.

References

[1] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "On accomodating mobile hosts in an integrated 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] 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.

[4] 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.

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

[6] Sunghyun Choi and Kang G. Shin, "Bandwidth Reservation in Mobile Cellular Networks Using ITS Navigation Systems," in Proc.of Intelligent Transportation Systems (ITSC'99), Tokyo, Japan, October 5-8, 1999. [

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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

[13] 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

[14] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Netw.,*vol. 1, pp, 329-343, June 1993

[15] F. P. Kelly, "Effective bandwidth of multi-class queues," *Queuing Systems,* Vol 9, pp 5-16, 1991

[16] J. Evans and D. Everitt, "Effective bandwidth based admission control for multiservice CDMA cellular networks", *IEEE Trans. Vehicular Tech.,* Vol 48, No 1, pp 36-46, January 1999.

[17] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "MRSVP, A resource reservation protocol for an integrated service network with mobile hosts," *The Journal of Wireless Networks,* Vol 7, No. 1, 2001.

[18] M. Naghshineh, A.S. Acampora, "QoS Provisioning in micro-cellular networks supporting multiple classed of traffic," *Journal of Wireless Networks,* No. 2, 1996.

[19] D. Hong and S. S. Rappaport, " Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized procedures," *IEEE Trans. on Vehicular Technology,* vol. 35, pp. 77-92, August 1986.

[20] M. M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns," *IEEE Journal on Selected Areas in Communication,* 15(7), pp. 1239-1252, September 1997.

[21] E. Jugl and H. Boche, " Analysis of Analytical Mobility Models with Respect to the Applicability for Handover Modeling and to the Estimation of Signaling Cost," in *Proc. ACM Mobicom '00,* Boston, August 2000.

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

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