Predictability of WLAN Mobility and its Effects on Bandwidth Provisioning

Libo Song[a], Udayan Deshpande[a], Ulaş C. Kozat[b], David Kotz[a], Ravi Jain[b,c]

[a]{lsong, udayan, dfk}@cs.dartmouth.edu, Computer Science Department, Dartmouth College, Hanover, NH, USA

[b]kozat@docomolabs-usa.com, DoCoMo USA Labs, San Jose, CA, USA

[c]Current email: ravijain@google.com

***Abstract*— Wireless local area networks (WLANs) are emerging as a popular technology for access to the Internet and enterprise networks. In the long term, the success of WLANs depends on services that support mobile network clients.**

Although other researchers have explored mobility prediction in hypothetical scenarios, evaluating their predictors analytically or with synthetic data, few studies have been able to evaluate their predictors with real user mobility data. As a first step towards filling this fundamental gap, we work with a large data set collected from the Dartmouth College campus-wide wireless network that hosts more than 500 access points and 6,000 users. Extending our earlier work that focuses on predicting the next-visited access point (i.e., location), in this work we explore the predictability of the time of user mobility. Indeed, our contributions are two-fold. First, we evaluate a series of predictors that reflect possible dependencies across time and space while benefiting from either individual or group mobility behaviors. Second, as a case study we examine voice applications and the use of handoff prediction for advance bandwidth reservation. Using application-specific performance metrics such as call drop and call block rates, we provide a picture of the potential gains of prediction.

Our results indicate that it is difficult to predict handoff time accurately, when applied to real campus WLAN data. However, the findings of our case study also suggest that application performance can be improved significantly even with predictors that are only moderately accurate. The gains depend on the applications' ability to use predictions and tolerate inaccurate predictions. In the case study, we combine the real mobility data with synthesized traffic data. The results show that intelligent prediction can lead to significant reductions in the rate at which active calls are dropped due to handoffs with marginal increments in the rate at which new calls are blocked.

I. Introduction

There are many challenges remaining before wireless local-area networks (WLANs) can effectively support voice, video, and other real-time interactive services. WLAN operators must incorporate mobile services and yet use resources efficiently to establish a profitable business model. To this end, the capability of predicting the time and place of a roaming user's next move can play a significant role.

The importance of mobility prediction to service provisioning has been widely acknowledged in the literature [1]–[13]. In most of these papers, voice is the main application of interest, and the objective is to reduce the likelihood of dropping roaming callers. These papers use mobility prediction for bandwidth reservation (or channel assignment) prior to the user's move from one cell to another (i.e. "handoff"). We summarize the related research and highlight their main assumptions in the following section. For now, it is important to note that most prior work have evaluated their predictors analytically or with synthetic data, leaving unanswered fundamental questions about how these predictors would fare with real user mobility patterns. To the best of our knowledge, published works based on real data are either limited to position tracking in cellular networks (e.g., [14]) or focuses on location prediction in WLANs independent of the time aspect (e.g., our earlier work [15] and others [16], [17]).

In this paper, we use real user mobility data to explore a range of predictor types to bring new insight to the challenge of predicting the *time* of a user's next move. Specifically, we use a two-month subset of the user mobility trace from the Dartmouth College wireless network [18], [19], which includes 545 access points and 6,181 users. The mobility trace specifies the association history of each user with the access points in the network. In this context, as a distinctive feature of the real data we use, mobility does not necessarily coincide with the change in user's position and it merely refers to the fact that user's device changed its access point. Note that for the application that we consider in our case study, bandwidth reservation for handoff VoIP calls, this change in registration is the relevant event as opposed to the user's geographical mobility.

One particular trend that we have observed in the literature is that predictors have been evaluated by the final outcome of the application performance. Although such evaluations can be valued as what finally matters, they do not clarify to what extent the performance gain is due to the quality of the prediction (i.e., low prediction error) or due to the accompanying control systems that use the predictions as their input and compensate for any prediction errors. Therefore, we divide our analysis into prediction and application stages, to measure the performance at the output of each stage separately. Nevertheless, evaluating predictors independent from the actual application is not a trivial matter. We need to identify a meaningful set of metrics that can properly evaluate predictors in terms of their ability to predict both the time and location of the next handoff. It is also true that different applications require different predictor outputs, for which the performance cannot be adequately measured by a single metric. In accordance with these observations, we defined a suite of metrics, which include: *accuracy*, *earliness-lateness*, and *under/over-provision*. These metrics are particularly useful for measuring binary decisions, continuous-valued outputs, and probability outputs, respectively.

This paper makes two major contributions. As our main contribution, we evaluate a series of predictors that can take advantage of various dependencies across time and space. This approach not only provides a sanity check for widely used mobility and prediction models in the current state of the art, but also derives insight about the predictor features that drive good performance in a real WLAN environment. Note that our main concern here is not on the complexity of the predictors and their implementation details; rather our goal is to understand the level of predictability for the real mobility traces that we have. With this in mind, our study is an important report on whether a campus WLAN environment is amenable to accurate prediction of handoff events.

As our second contribution, we present a case study that quantifies the possible gains due to mobility prediction in an actual application. In our case study, we focus on VoIP as the application of interest and we evaluate the use of mobility predictors for advance bandwidth reservation to maintain VoIP service quality after handoffs. We measure the performance using application-specific *call drop rate* and *call block rate* metrics. This approach allows us to dissect the performance gains due to the quality of predictions and the reservation policy. We focus on the influence on call drop rate and call block rate when mobility prediction applies to a variety of call admission schemes. We left out the technical details for implementing different bandwidth-reservation and call-admission systems.

Our results reveal a range of predictability for time and location, according to the proposed metrics. Although some predictors do better than the others on average, we also observe that certain prediction techniques work much better for some users or at particular locations. In general, however, a clear outcome of our study indicates that accurate prediction with fine time granularity requires more sophisticated and expensive data collection technologies than simply relying on the user associations. It is promising, however, to see that even with a modest level of predictability, our case study indicates that the intelligent use of predictor outputs can improve the performance of some applications significantly. More specifically, our case study, which superimposes a synthesized traffic data over the real mobility data, shows that with prediction, we achieve significant reductions in call drops while minimally affecting the call block rates.

The rest of the paper is organized as follows. In Section II, we present a brief overview of the related literature. Section III provides a model outlining our assumptions about the system and the data at hand. In Section IV, we introduce the specifics of predictors we examined. Section V describes the proposed performance metrics and evaluation method. Section VI explains and discusses the main results on predictors. Then, in Section VII, we introduce our case study following standard evaluation methods that are widely used in the literature. Finally, Section VIII concludes the paper by summarizing our findings and discussing ongoing work.

II. Literature Overview

Location prediction is the process of predicting a roaming user's next location given their current location and prior movement history; in this paper we focus on predicting the next cell that a mobile user will visit. Many location-prediction techniques exist [20], and we evaluated several with real data [15]. For many applications, however, it is important to predict *when* the user is likely to move, as well; in this section we summarize research that attempts joint time and location prediction, especially in the context of resource reservation designed to improve telephony quality of service. Here, we emphasize two points: (i) prediction accuracy largely depends on the granularity of the available data set and (ii) the success of a predictor strongly depends on how close its assumptions about the user mobility models the real situation. In the following, we use the terms *location*, *cell*, and *access point (AP)* interchangeably, and *position* should be understood as the point in space.

Every predictor is limited by the granularity of the available information. There are three common options.

1) Record the visited locations and time of handoff to these locations for each user. Since this information is readily available to every network operator that supports roaming, it is widely used [1], [2], [4], [6], [11]. The data we use in our evaluations also falls into this category.
2) Record the position and velocity of each user. This information can be obtained by signal triangulation techniques or client-carried GPS devices, both increasingly common in cellular phones. Here, mobility prediction requires algorithms that can infer cell boundaries and user velocity [4], [7], [9], [10].
3) Use high-level information that describe the distinguishing, designating or limiting properties of each cell, derived from road and building maps [1], [12].

Every predictor also assumes, implicitly or explicitly, certain characteristics of user behavior. Its success depends on the degree to which user behavior actually fits these assumptions, and in the variance of users' behaviors. Unfortunately, little of the prior research evaluates mobility predictors using data about real user mobility. To study the real users' behaviors, in our early work [15] we evaluated location predictors using the Dartmouth WLAN user traces. In this paper, we use data from the same campus WLAN to evaluate time-and-location predictors directly, and in the context of a bandwidth-reservation scheme for roaming VoIP callers.

The rest of this section presents specific examples from the prior art.

In one of the earliest works specifically targeting location prediction in WLANs, Lu and Bharghavan [1] suggest the use of location semantics (such as office space, corridor, or lounge), and their relevance to the individual users, to guide location prediction. For instance, if a specific location is known to be an office space, the residents of the office are likely to be moving towards this location provided that they are approaching this location. If the current location is a lounge, the aggregate history of all users that have used that particular lounge should give a rough idea about where the next location should be. This quite intuitive approach may be misleading, because it is often the case that there is not a one to one relation among physical space and access point. A physical space such as office and lounge can be served by more than one access point. Similarly, an access point can cover many physical spaces. Both instances violate the one-to-one assumption in Lu and Bharghavan [1].

Others have proposed to use similar high-level information for cellular networks, where vehicular speeds and handoffs are common. For instance, some argue that road maps, which show road segments and their intersections, can help estimate the path of a caller in a vehicle and thus predict the time and place of the next handoff [12]. Rather than modeling the transitions between cells, they model the transitions between road segments. Using precise velocity estimates, they are able to obtain the most likely handoff events and corresponding handoff times. The transitions between the road segments follow a group mobility pattern, used by Choi and Shin in the context of transition probabilities between the cells [6].

Choi and Shin [6] further advocate a model in which some handoffs are driven by external factors, and less by individual history. Therefore, the handoff probability can be estimated by the aggregate of all users' history at each location. Indeed, their handoff probability is conditioned on the current location, the previous location, and on the time already spent in the current location. Furthermore, they suggest that predictions can be further improved if they are conditioned on the time of the day (i.e., busy hours vs. normal hours) as well as on the specific day (i.e., weekday vs. weekends).

Levine et al. [2] use the shadow cluster concept. Depending on the velocity of the user, its current base station determines a set of neighboring cells that can be visited by the mobile. The past residence history, call duration statistics, neighborhood relations, and direction of the user are all used to estimate handoff probability at a future time from the current cell of an active call to all other cells in the shadow cluster.

Other research proposes to measure and use the signal to interference and noise ratio, or GPS positioning [4], [7], [9]. One [4] considers a 2-level mobility estimation by performing pattern matching predictions for inter-cell handoffs and modified Kalman filters for intra-cell behavior. Another [7] uses a recursive least-square method to predict the next cell using location inference obtained by a fuzzy logic. Another [9], on the other hand, uses a direction-estimation-based method to form a most likelihood cluster (MLC) of future cells. Interestingly, these authors also distinguish prediction-conforming and non-conforming users, and later penalize non-conforming users in the resource-allocation phase. In all these models, the accuracy of handoff prediction depends on how well the position, velocity, and acceleration are estimated. Some recent works report promising results both in urban and suburban area measurements at vehicle and pedestrian speeds by using a first-order autoregressive model for state transitions and Kalman filtering for current state estimation [14], [21]. Robust location tracking with mobile base stations is also among the investigated scenarios [22].

Pattern-matching techniques constitute another important class of techniques for predicting either location or time. These techniques generally assume a mixture of different Markovian models as the source of movement patterns and some of them are proven to be asymptotically optimal for such patterns [11], [23], [24]. In an earlier experimental comparison of these and other location-prediction methods [15], we found that a simple Markov predictor obtains the best prediction accuracy with less complexity and using less space than the more sophisticated predictors.

III. System Model

Before we describe the predictors we study, and our method for evaluating them, we first outline our assumptions about the users, the data available to predictors, and the system for making and using predictions.

We envision a wireless network in which users associate with one access point (or cell) at a time, and roam from cell to cell as needed to remain connected while they move. Indeed, it is possible that the user's device may switch cells (i.e., re-associate with a different access point) without physically moving; in WiFi networks, some stationary clients have been known to alternate rapidly between two or three access points when no one signal clearly dominates the others [18]. Occasionally, the user leaves the network by switching off their device or by moving out of range. This sequence of handoffs for each user, indicating the time and access point of each move (or "OFF" when the user leaves the network), is their movement history.

We assume that there is a centralized or distributed mechanism in place for (i) collecting the history, (ii) performing online predictions, and (iii) distributing prediction results to relevant application agents. The challenge of designing scalable yet optimal resource allocation that relies on online measurements is inherent to all distributed systems and is beyond the scope of this paper.

IV. Predictors

We begin by describing predictors that can predict the time of the next handoff, which can be used in tandem with a predictor that predicts the next location (of which many have been proposed in the literature). We then consider integrated approaches that predict the location and time jointly.

A predictor examines a movement history, which (by our afore-mentioned assumption) is a sequence of cell associations (access point, time), rather than a record of client position and velocity. Thus we consider predictors that consider location symbolically rather than geometrically.

A. Predictors

We consider three fundamentally different types of predictor: Markov predictors, moving-average predictors, and CDF predictors. Some of these can be used to predict the handoff time only, while some can predict both time and destination. Each of these predictors can be trained on either a) the history of movements by a single individual, or b) the history of movements by all users. Whether trained with "individual" or "aggregate" data, the predictor is used in the same way, to predict each handoff of each user. In addition to these three types of predictors, we also include a Static Neighbor Graph predictor for comparison.

First, we define the three basic predictor types followed by the Neighbor predictor.

1) Markov predictor: The order-k (or "$O(k)$") Markov predictor takes a sequence of symbols $(a_1, a_2, \ldots, a_n)$ as a history string, and tries to predict the next symbol from the current *context*, that is, the sequence of the k most recent symbols in the history $(a_{n-k+1}, \ldots, a_n)$.

Consider history $H = a_1 a_2 \ldots a_n$, and let substring $H(i, j) = a_i a_{i+1} \ldots a_j$ for any $1 \le i \le j \le n$. Let X be a random variable symbol. Let $X(i,j)$ be a string $X_i X_{i+1} \dots X_j$ representing the sequence of discrete random variates $X_i, X_{i+1}, \dots X_j$ for any $1 \le i \le j \le n$. Define the context $c = H(n-k+1, n)$. Let $\mathcal{A}$ be the set of all possible symbols. If X has order-k stationary Markov distribution, for all $a \in \mathcal{A}$ and $i \in \{1, \dots, n-k\}$, its distribution satisfies

$$\begin{aligned} &P(X_{n+1} = a | X(1,n) = H) \\ &\quad = P(X_{n+1} = a | X(n-k+1, n) = c) \\ &\quad = P(X_{i+k+1} = a | X(i+1, i+k) = c). \end{aligned}$$

At any given time, we can estimate the transition probability to a using current history H and current context c of most recent k symbols as

$$P(X_{n+1} = a|H) \approx \widehat{P}(X_{n+1} = a|H) = \frac{N(ca, H)}{N(c, H)}, \quad (1)$$

where $N(s', s)$ denotes the number of times the substring s' occurs in the string s. The Markov predictor returns the most likely next symbol as its output:

$$X_{n+1} = \text{argmax}_{a \in \mathcal{A}}(P(X_{n+1} = a)). \quad (2)$$

We further define the $O(k)$ *fallback* Markov predictor, which falls back to an $O(k-1)$ Markov predictor whenever the $O(k)$ predictor is unable to make a prediction, which occurs whenever $N(c, H)$ is zero, that is, whenever the current context has never been seen before. We also define the "order-0" Markov predictor, which always outputs the symbol that occurs most frequently in the history H. Our earlier work [15] shows that $O(2)$ Markov with fallback is one of the best location predictors; in that work, a history H represented a series of locations.

2) Moving-average predictor: Moving averages are commonly used to predict a trend in a sequence of values. The order-k (or "$O(k)$") moving-average predictor takes a sequence of values and predicts that the next value of the sequence is the average of the last k values in the sequence.

Consider a history H of values $v_1, v_2, \dots, v_n$. The order-k moving-average predictor estimates the next value to be

$$v_{n+1} = \frac{1}{k} \sum_{i=1}^{k} v_{n-i+1}. \quad (3)$$

3) CDF predictor: Rather than attempting to predict the next symbol or value in a sequence, this predictor works with inequalities. Specifically, it produces the probability that the next value is less than (or greater than) a given value. It does so by computing the observed cumulative distribution function (CDF) of the historic values, and using the CDF to measure the probability of a given value appearing in the distribution.

Consider a history H of values $v_1, v_2, \cdots, v_n$. Suppose V is the random variate, which outputs the actual values in H, and P is its distribution. The CDF predictor computes the observed CDF function of V from the histogram, that is,

$$\text{CDF}(V < v) = \frac{1}{n} \sum_{i=1}^{n} I(v_i < v), \quad (4)$$

where I is the indicator function. In a similar fashion, we can compute the probability of values occurring in range $a \le V < b$, by simply computing $P(a \le V < b) \approx (\text{CDF}(V < b) - \text{CDF}(V < a))$.

4) Static Neighbor graph predictor: We introduce a simple "straw-man" predictor, the Static Neighbor Graph Predictor, to compare with our predictors.

Using users' current neighbor locations as the prediction is an obvious way to predict future locations. Mishra et al. [13] present an algorithm to dynamically build a user's neighbor graph to cache context for fast handoffs. If the network topology does not change quickly over time, a location predictor can use pre-collected AP topology information to predict the user's mobility. In our experiment we use one month's data to construct a directed graph representing transition history as follows: when we observe a user move from an AP i to another AP j, if the edge (i,j) is not in the graph, we will add the directed edge to the graph and set the weight of this edge to 1; if the edge (i,j) is in the graph, we add 1 to the weight of the edge. At the end of the month, we have a directed graph with weighted edges and normalize the weights so that $\forall i, \sum_j w_{ij} = 1$, where w_{ij} is the weight on edge (i,j). When a prediction is requested, the predictor finds the user's current location i in the graph, and returns a list $\{(j, w_{ij})\}$ for all edges (i,j) originating at i.

B. Predictor uses

We now show how to apply the first three techniques to predict the duration of the current stay and the likelihood of handoff to any other access point. Some applications need an estimate of the handoff time (or, equivalently, the duration of the stay at the current location); other applications are more concerned about knowing whether the handoff will happen "soon" and if so, the destination of the handoff.

1) Duration prediction: To predict the time of handoff, one can look for patterns in the residence times rather than the absolute time of the handoff. We apply all three predictors, i.e. Markov, moving-average, and CDF to predict the next duration.

We use the Markov predictor by considering the history $H = d_1, d_2, \dots, d_n$ of previous durations, again quantized into intervals of size Δt. Each quantized duration bucket is thus a symbol in $\mathcal{A}$.

We use the moving-average predictor as follows. Consider again the history $H = d_1, d_2, \dots, d_n$ of previous durations; there is no need to quantize the times as we did above. Then by (3) the order-k moving-average estimate of the residence time at the current location is

$$d_{n+1} = \frac{1}{k} \sum_{i=1}^{k} d_{n-i+1}. \quad (5)$$

Similarly, we do not need to quantize the duration to apply CDF predictor. Consider again history $H = d_1, d_2, \dots, d_n$ of previous durations, we estimate a value of d such that with probability p the duration will be less than d. We define the CDF by counting the fraction of durations shorter than a given time:

$$CDF(t) = \frac{1}{n} \sum_{i=1}^{n} I(d_i \le t),$$

where $I()$ is the indicator function. We can interpolate to predict that the user will stay shorter than $d = (t_l + t_h)/2$, where t_l is the minimum t that satisfies $CDF(t) \geq p$, and t_h is the maximum t that satisfies $CDF(t) \leq p$. The predictor parameter p expresses the desired confidence in the result.

We can apply each predictor to the history observed in any location as well as to the history observed only at the current location to make them "location-independent" and "location-dependent" predictors respectively. We can also apply the predictors to each user's history (individual predictors) or apply them to all users' history (aggregate predictors).

2) Joint Location and Time Prediction: Many applications need to predict both the time and destination of the next handoff; our telephony case study in Section VII is one example. In this case the goal is to predict, for every destination, the probability that a handoff will occur within the next Δt period, conditioned on the current location and the duration of the visit so far. Our approach is to combine the Markov location predictor and the CDF time predictor.

Consider a user's movement history $H = (t_1, a_1), (t_2, a_2), \ldots, (t_n, a_n)$, in which t_i is the time that the user arrived at location a_i. From H we extract the location history $L = a_1, a_2, \ldots, a_n$, and from L the order-k location context $c = L(n-k+1, n) = a_{n-k+1}, \ldots, a_{n-1}, a_n$. We then search the history L for instances of the context c. We need to examine the destinations that follow each such instance, and in particular to examine the duration of the visit preceding each of those destinations, to be able to predict the duration in the current context. So, we need to extract the set of durations for each possible destination x:

$$D_x = \{d_i \,|\, d_i = t_{i+1} - t_i \,\text{where}\, L(i-k+1, i+1) = cx\} \quad (6)$$

From each duration set D_x, we use the CDF predictor in Section IV-A.3 to compute the conditional probability $P_x(t \leq d < t + \Delta t \,|\, c, t)$ that the user will move to location x within Δt seconds after the current elapsed residence time t. We can also use the $O(k)$ Markov predictor to compute the probability $P(x)$ of every possible next location x with Equation (1).

Therefore, the probability of the user moving to each of the possible locations x within the next Δt seconds, given the current context c at the current elapsed residence time t, is

$$P(x \,|\, c, t) = P(x) \cdot P_x(t \leq d < t + \Delta t \,|\, c, t) \quad (7)$$

We name this predictor MarkovCDF.

This predictor is always conditioned on the current location context. As described above, the predictor builds per-user tables, but it is equally possible to build aggregate tables from all users' movement histories.

V. Performance Metrics

A proper evaluation of predictors requires meaningful performance metrics. In the context of a given application, an evaluator can use performance metrics relevant to that application. For instance, if the application is voice telephony, the rate of dropped calls emerges as a natural choice. Application-specific metrics are often affected by factors other than prediction quality, such as the bandwidth-reservation policy in a voice telephony application, so the metric does not provide direct insight into the quality of the predictor. Furthermore, it is valuable to develop generic metrics to allow the study and comparison of predictors outside the context of particular applications. In this section, we develop three metrics with a focus on time or time/location prediction: accuracy, earliness/lateness, and over/under-provisioning. In Section VII we identify additional application-specific metrics for our case study.

A. Accuracy

In our previous work [15], we evaluate several location predictors using the metric *accuracy*, defined as the ratio between the number of correct predictions and the number of all predictions. A prediction was correct if it predicted the next location correctly. We quantized time and used the accuracy metric for some of our predictors.

B. Earliness and Lateness

Accuracy is a good metric for applications that require hard decisions, e.g., whether the handoff will occur or not in a given time interval, from the predictors. In many cases however it is more desirable to measure how much the predicted handoff time differs from the actual handoff time. In most applications, overshooting or undershooting the actual time of handoff will have different implications; for example, a late request to reserve bandwidth for a roaming voice call will be useless, whereas an overly early request will consume excess resources.

Hence, we define separate metrics; when a handoff occurs at time t_h, and the handoff was predicted to occur at time t_p, the prediction *earliness* is $t_h - t_p$ if $t_h \geq t_p$, and *lateness* is $t_p - t_h$ if $t_h < t_p$. We compute each user's average earliness across all movements in which the prediction was early, and each user's average lateness across all movements in which the prediction was late.

C. Over-provision and Under-provision

When a predictor is used for advance resource allocation, an incorrect prediction can lead the system to reserve too many, or too few, resources. We consider predictors, such as the CDF predictors defined in the preceding section, that provide the probability of handoff for each possible destination, and imagine a system that provisions resources at each destination in proportion to these probabilities. When the handoff occurs, we can determine the amount that resources were under-provisioned at the actual destination, and over-provisioned at all other locations.

We can now precisely define under- and over-provisioning.

Under-provisioning: At any time t that the predictor indicates that the user u may move from location i to location j with probability $P(j|u, i, t)$, and the handoff occurs, we measure the amount of under-provisioning for this handoff as

$$E(u, i, j, t) = 1 - P(j|u, i, t).$$

In effect, under-provisioning is equivalent to computing the error with respect to an "ideal predictor" that always returns the correct prediction: probability 1 for the correct destination just before the handoff is to occur, and probability 0 for other destinations or other times. Thus, a lower value of E represents better prediction quality. When a predictor is used periodically, as might the CDF predictors defined in the preceding section, under-provisioning applies only to the last prediction before the handoff.

Over-provisioning: Any time t that the predictor indicates that the user u may move from location i to location j with probability $P(j|u,i,t)$, and a system provisions resources in accordance with that probability, the resources are wasted if the handoff does not occur as predicted. We define the amount of over-provision as

$$F(u,i,j,t) = P(j|u,i,t),$$

because an ideal predictor would have produced probability zero. Thus, a lower value of F represents better prediction quality. When a predictor is used periodically, we can compute the over-provision metric for each such prediction, under the rationale that a system using such a predictor would waste resources at predicted destinations whenever a handoff does not occur, or at all locations other than the actual destination whenever a handoff does occur, within that prediction interval.

As a more comprehensive representation of these metrics, we compute per-user and per-AP averages; a non-uniform distribution would mean that some users or some locations were more amenable to prediction than the rest. We define the average under-provision per user as

$$E(u) = \frac{\sum_{i,j,t} E(u,i,j,t)}{\sum_{i,j,t} 1},$$

and the average under-provision per access point as

$$E(j) = \frac{\sum_{u,i,t} E(u,i,j,t)}{\sum_{u,i,t} 1}.$$

To compute the average per-user and per-AP over-provisioning, we must pay special attention to how we normalize the total error. For under-provisioning, measured only at the destination and only at a handoff, the denominator is clearly the number of handoffs. For over-provisioning, which is measured at every access point and at every prediction attempt, the denominator could be as large as the number of access points times the number of prediction attempts. This approach underestimates the error, however, in large networks where most transitions $i \to j$ simply never occur and any reasonable predictor produces probability zero for those cases. We compensate by summing over an indicator function $g(i,j)$, which represents, for each pair of access points i and j, whether any user has ever moved from AP i to AP j. That is, $g(i,j) = 1$ if some user's history includes a move from AP i to AP j, and $g(i,j) = 0$ otherwise.

Accordingly, we define the average over-provision per user as

$$F(u) = \frac{\sum_{i,j,t} F(u,i,j,t)}{\sum_{i,j,t} g(i,j)},$$

and the average over-provision per access point as

$$F(j) = \frac{\sum_{u,i,t} F(u,i,j,t)}{\sum_{i,j,t} g(i,j)}.$$

In last two sections, we describe the metrics that we used, the predictors that we evaluated, and the predictors' uses. Table I provides a summary of our discussion so far. We show the evaluation results of the predictors according to theese uses in the following section.

VI. Performance Evaluation

In this section we evaluate the performance of several predictors using the metrics in the preceding section. First, we describe the Dartmouth mobility data we used in our evaluation.

A. Evaluation data

The mobility data used in this paper was collected on Dartmouth's campus-wide WiFi network, which provides 11 Mbps coverage to the entire campus through over 500 access points. Although outdoor coverage is not guaranteed, due to the compactness of the campus, the interior APs tend to cover most outdoor spaces. A typical day of activity includes between 2,500 and 3,500 users. The mobility data does not reveal the actual position of the wireless user, but rather the identity of the access point (AP) that serves the user at that moment. In some situations, the user's device will roam from AP to AP even when the user does not physically move; the *location* may change although the *position* did not. Hence, our evaluations are limited to the predictor types that do not rely on more than such location history. The collection and processing of the data is described more fully elsewhere [15].

In this data set, each user's trace is a series of times and locations; the location is the name of an access point visited at that time. The timestamp granularity is one second. The special location "OFF" represents the user's departure from the network, which occurs when the user turns off their computer or their wireless card, or moves out of range of all access points. In our experiments, we do not compute the metrics for transitions to or from location "OFF."

We chose two months of data collected during fall of 2003 for our experiments; specifically, October and November 2003. We chose these two months because the data was nearly complete (some other months are missing brief periods) and longer periods are computationally expensive to simulate.

Our predictors process users' traces in an online fashion, updating internal tables and predicting each move where possible. We compute our metrics only during the second month, using the first month to seed the tables.

B. Results

We performed an extensive set of experiments to evaluate our predictors with the aforementioned metrics. For conciseness, we highlight our major findings using a smaller subset of these experimental results.

1) Duration prediction: Intuition suggests that people tend to stay at APs in specific patterns. One specific behavior we expected was a similarity in the residence times. We first experimented with Markov predictors to search for stationary patterns in the user history. Note that, to conduct Markov predictions, we needed to quantize the durations as symbols. Figure 1 explores the performance sensitivity against different orders of dependency on the previous durations as well as the sensitivity to the quantization interval. The results demonstrate that a 10-minute interval was about 20% less accurate than a one-hour interval for the same order Markov predictor. The rightmost curve, i.e., the O(2) Markov predictor with fallback using a one-hour interval, is the best with a median value of 0.88. The worst is the O(2)

TABLE I
PREDICTOR TYPES, USAGE, AND METRICS

predictor		Markov	Moving Average	CDF	MarkovCDF	Neighbor
usage	duration	yes	yes	yes	no	no
	integrated	no	no	no	yes	no
metrics	accuracy	yes	yes	yes	no	no
	earliness/lateness	yes	yes	yes	no	no
	under/over provision	no	no	no	yes	yes
dependency	location†	yes/no	yes/no	yes	yes	yes
	time‡	no	no	yes	yes	no
profile	individual	yes	yes	yes	yes	no
	aggregate	yes	no	yes	yes	yes

†The location dependence means that the prediction depends on current and/or more previous visited locations.
‡Time dependency means that the prediction depends on how long a user has been at the current location.

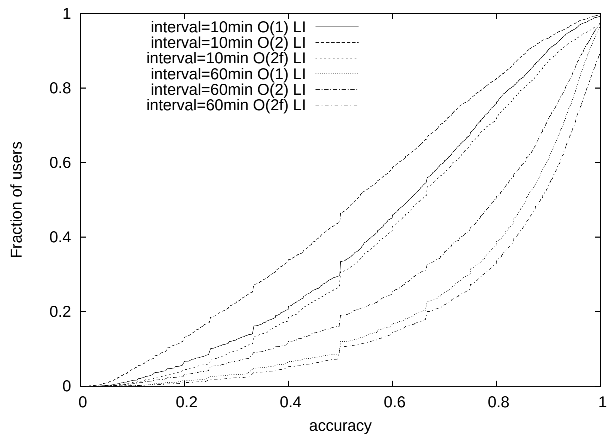

Fig. 1. Duration prediction with per-user based location-independent (LI) Markov predictor. The duration is quantized using one-hour intervals and 10-minute intervals. The notation $O(2f)$ means $O(2)$ Markov with fallback.

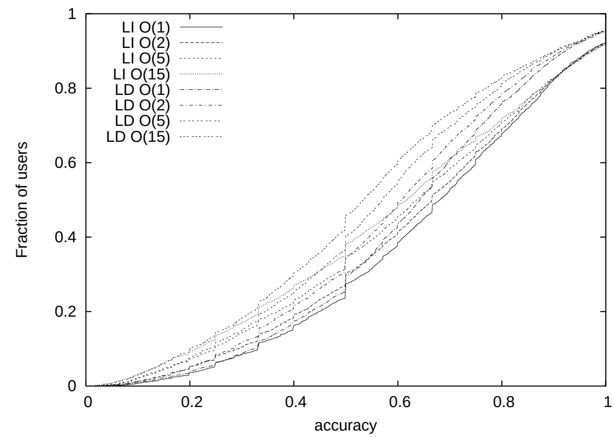

Fig. 2. Duration prediction with location-independent (LI) or location-dependent (LD) Moving Average predictors. The prediction output is quantized into one-hour intervals to compare with the actual duration.

Markov predictor without fallback using 10-minute interval, with a median of 0.53. It is not surprising that the bigger interval led to higher accuracy, because there was a higher probability that the actual move falls within the interval. Moreover, with a bigger interval the predictor had fewer states, reducing the chance of failure. The lagging performance of O(2) with no fallback is mainly due to the fact that we count the cases where predictor cannot find a similar context before as unsuccessful predictions. Yet, the fallback results suggest that whenever there is sufficient context, O(2) provides better accuracy than O(1). The location-dependent Markov duration predictors were less accurate than the location-independent Markov predictors due to lack of sufficient data to train the predictors.

Our experiments with Moving Average and CDF predictors showed that Moving Average had better performance; Figure 2 shows its results. Since Moving Average predictor can operate on continuous variables, we do not need to quantize the input to the predictor. However, the accuracy metric itself requires that we quantize the output of the predictor. With quantization interval of one hour, we find out that the median accuracy varied from 85% to 90% for the experimented prediction orders and dependencies. Figure 2 explores different orders of Moving Average predictors as well as the location dependence. Although the curves are close to each other, the accuracies of the location-dependent Moving Average predictors are slightly worse than the results of the location-independent predictors. For both cases using a one-hour interval, the curves of the results cross over and are close. However, when using a 10-minute interval (not shown), the higher-order predictors perform better than lower-order ones; whereas, the location-independent predictors still are better than location-dependent ones.

Although accuracy is a sensible measure for discrete input/output predictors like the Markov predictor, it is not as suitable for continuous input/output predictors. Many applications want to know how predictions deviated from the actual value. Therefore, we used the earliness and lateness metrics to measure the error of the duration prediction.

Figure 3 compares three predictors in terms of the *earliness* using each of their best observed results within a set of parameters that we explored. Clearly, the Markov duration predictor was superior. The CDF duration predictor was slightly less accurate than the Moving Average duration predictor. Nonetheless, all of these predictors failed to predict duration accurately for many users. The median user experienced an average earliness of 604, 1156, and 1218 seconds, for location-dependent $O(2)$ Markov duration predictor using 10 minutes interval, $O(1)$ location-independent Moving Average predictor, and CDF duration predictor using probability 0.1, respectively. The 80% percentile user has an average earliness of 1662, 1690, and 4479 seconds, for the above three predictors, respectively.

Due to limited space we do not show the *lateness* results, but we found even more skewed distributions and high mean lateness. The median user had mean lateness of about 1000 to $10,000$ seconds using the same set of predictors depicted in Figure 3. Notably, the Markov predictor was far worse than the Moving Average predictor, because there were relatively few contexts with high residence times, leading to wildly inaccurate predictions when durations were large.

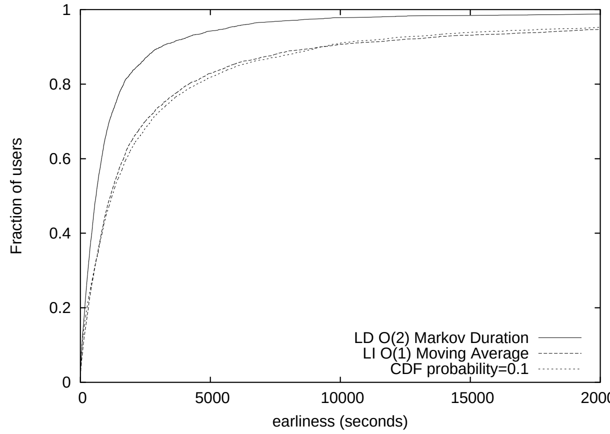

Fig. 3. Comparison of earliness measurement of predictions: Markov duration prediction, CDF duration prediction, and Moving Average duration prediction. The curves in the plot are the best observed curves of each of their predictors, across a set of parameters we explored. We truncated at 20,000 seconds.

The accuracy, earliness, and lateness metrics indicated that a precise time prediction is not a realistic goal over our real WLAN traces. This fact motivated us to investigate predictors with soft outputs that represent the likelihood of the events rather than their exact values, e.g., "what is the likelihood of leaving the current location within 5 minutes?" Such likelihoods in time prediction may depend on how long the user has stayed in his current location as well as the past history and predicted future location. Since the accuracy, earliness, and lateness metrics are not suitable to measure the quality of such predictions, we use the under-provision and over-provision metrics to quantify the errors in probabilities. In what follows, we present our results for joint time and location prediction, i.e., the MarkovCDF predictor, using under/over-provision metrics.

2) Joint Time and Location Prediction: For joint time and location prediction, the MarkovCDF predictor returns a list of possible next locations with the probability for each location, to which the user may move within a specified time. In Figures 4-7, we show the under-provisioning and over-provisioning per user and per AP, where the latter helps us to understand whether the predictability is an attribute of the location or of the user. Results reflect the performance differences for different prediction orders and for the cases when individual or aggregate histories are used. In these experiments, the predictions were updated at the end of every 60-second time slots, and the handoff probability in the next 60 seconds was computed. As a comparison, we also show the results of the Neighbor predictor in these plots.

For both under-provision and over-provision, lower values (upper curves) represent better performance. Figure 4 shows the CDF plots of average under-provision for users and Figure 5 shows the CDF plots of average under-provision for APs. The aggregate-profile predictors used all users' history information so that if a particular user did not visit an AP before, the predictor could still make a prediction based on the movements of other users at that AP. The probabilities spread out over many locations, perhaps far more than a single user usually visited. Furthermore, the probabilities depended on how long the user had stayed at the current location. Thus, for a given time, the predictor might predict the user would move with

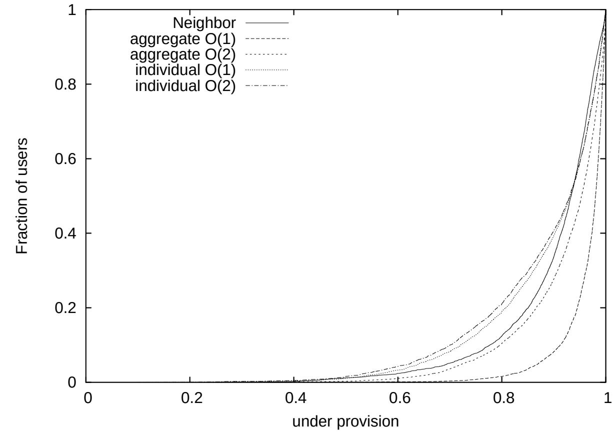

Fig. 4. Average under provision for users. Neighbor is the Static Neighbor Graph predictor. All the others are the MarkovCDF predictors. Aggregate table and per-user table are compared. Lower "under provision" numbers are better.

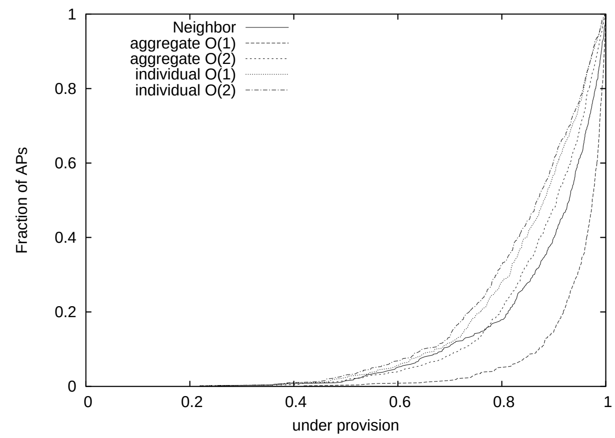

Fig. 5. Average under provision for APs. Neighbor is the Static Neighbor Graph predictor. All the others are the MarkovCDF predictors. Aggregate table and per-user table are compared. Lower "under provision" numbers are better.

a low probability. The probabilities might also be skewed by some extreme users who bounce back and forth among several APs. Therefore, the provision at the location that the user actually moved to was too little. In contrast, the individual-profile predictor returned probabilities that reflected the user's personal mobility pattern so that the user was better provisioned.

The Neighbor predictor can predict a location that the user had not visited before but it did not consider how long the user had stayed in the location. It always expected the user to move within the given period. As a result the Neighbor predictor made higher under-provision than the individual-profile MarkovCDF predictors. Although the Neighbor predictor beats the aggregate-profile MarkovCDF predictors for most users, the O(2) aggregate-profile MarkovCDF predictors made lower under-provision for most APs than the Neighbor predictor.

Within each group of individual or aggregate profile MarkovCDF predictors, the higher order predictors made lower under-provisions because they made more accurate predictions.

Figures 6 and 7 show the over-provision metrics. Clearly, the Neighbor predictor far over-provided due to its assumption that the user will always leave within the period. The individual-profile MarkovCDF predictors still fared better than the aggregate profile predictors.

Both the under-provision and the over-provision were com-

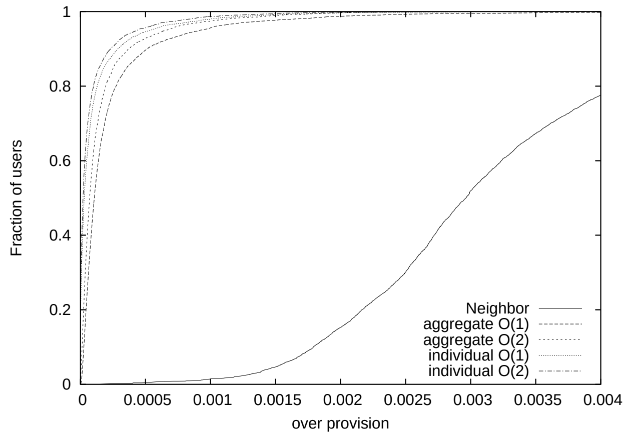

Fig. 6. Average over provision for users. Neighbor is the Static Neighbor Graph predictor. All the others are the MarkovCDF predictors. Aggregate table and per-user table are compared. Lower "over provision" numbers are better.

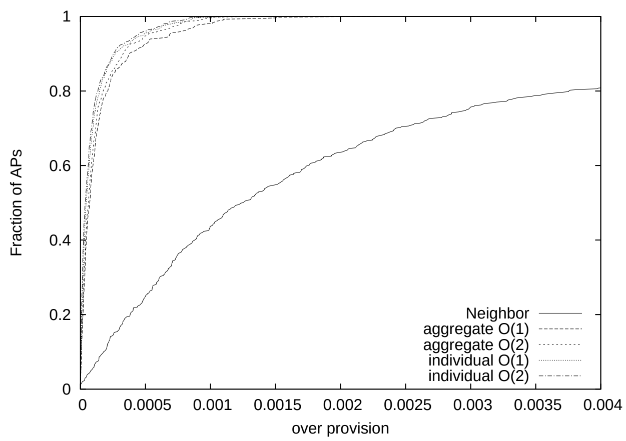

Fig. 7. Average over provision for APs. Neighbor is the Static Neighbor Graph predictor. All the others are the MarkovCDF predictors. Aggregate table and per-user table are compared. Lower "over provision" numbers are better.

puted by accumulating the absolute errors of individual user's predictions. For our data set, however, there were more than 2000 users during each day. With some reservation policies, an individual user's over-provision may benefit other users who are under-provided by their own prediction. To explore the overall performance of the integrated location and time predictors, we describe our VoIP case study in the next section.

VII. Case Study

The results of the previous section indicate that it is in general difficult to obtain very accurate predictions for handoff times by relying only on the AP association history of WLAN clients. In this section, we demonstrate this point by working with a voice-over-IP (VoIP) application. This application is a natural choice, considering the increasing popularity of VoIP and related multimedia applications on WLANs. Furthermore, voice has been extensively studied in the context of cellular networks and this section complements those studies. François and Leduc studied mobility prediction's influence on QoS in wireless networks by examining call-block rate and call-drop rate with arbitrary prediction accuracies [25]. In this paper, we study the influence using the prediction results from real traces. Another study [17], also based on the Dartmouth data set, attempts to use location prediction to improve handoff latency in VoIP calls in the context of Mobile IP. Improvement of handoff latency is outside the scope of this paper. Instead, we focus on using mobility prediction to reserve bandwidth for handoff calls. The technology for implementing these reservations in various networks is also beyond the scope of this paper.

The Dartmouth dataset contains both laptops and VoIP handsets. There are about 150 PDAs and VoIP-dedicated devices. Those VoIP devices only generate a small fraction of the total traffic, but we see VoIP as emerging and increasing use of the Dartmouth WLAN [19]. Increasingly, though, laptops are also used to make VoIP calls. Although laptops are physically stationary while making calls, they may sometimes roam to other APs. Hence it is useful to make reservations at APs for laptops as well. We expect that as the proportion of VoIP devices increases on campus, the performance of our predictors will change, and we plan to study those trends in future experiments.

In our scenario, the wireless network is capable of supporting roaming telephone users. When a user has an ongoing call and moves from one access point (AP) to another, we refer to that call as a *handoff call*. When a user initiates a call while associated with an AP, we refer to that call as a *new call*. All calls require dedicated bandwidth at their current AP. If the AP lacks the bandwidth for a new or handoff call, the call fails: a failed handoff call is a "call drop" and a failed new call is a "call block". The literature often assumes that call drops are much more frustrating to users than call blocks, so the goal of mobility prediction in this particular application is to reserve bandwidth, in advance of handoffs, to reduce call drops at the expense of a small increase in call blocks. Specifically, we define the drop rate and the block rate

$$DR = \frac{\text{number of dropped calls}}{\text{number of attempted call handoffs}},$$

$$BR = \frac{\text{number of blocked calls}}{\text{number of attempted calls}}.$$

Note the difference in the denominator of each equation. *DR* is not normalized by the total number of calls, or even attempted calls, but by the actual number of call handoffs, because every handoff is an opportunity for a call drop, whereas only call attempts are opportunities for call blocks.

We constructed a simulator to experiment with this scenario. We drove user mobility within the simulator with the Dartmouth mobility data, using the same two-month long subset as in the preceding section. We generated a synthetic calling pattern to model the hypothetical telephony behavior of these users.

We make the following assumptions about users and calls:

1) All calls require the same, fixed amount of bandwidth.
2) The bandwidth of an AP is fixed and the entire capacity of every AP is used for VoIP calls. (Or, a fixed capacity is reserved for VoIP calls.)
3) The duration of a call is exponentially distributed with mean λ_d.
4) The length of each user's inter-call time (between the end of one call and the beginning of the next call) is exponentially distributed with mean λ_i.
5) The calling behavior of every user on the campus shares the same model and parameters.

6) When a user with an onging call goes OFF, the call will be terminated gracefully (not counted as a call drop). When a user is in the OFF state, no calls will be initiated from this user.

A. Bandwidth Reservation Algorithm; Call Admission Control

Our predictor evaluations show that when predictors return an exact handoff time as the output, the results are highly skewed, and making distinct reservations for each user becomes impractical. Thus, we turn our attention to predictors that can return soft values, i.e., handoff probabilities, and design reservation mechanisms that dynamically allocate bandwidth for all handoff calls using these soft values. Such a reservation policy benefits from the *law of large numbers* provided that there are enough number of users and a useful sense of stationarity exists in the mobility patterns. Since the moving average predictor cannot provide such soft values and the Markov predictor is (in effect) a special case of the CDF predictor, we use only the CDF predictor from the first part of the paper. We compare the simulation results when the CDF predictor is used for predictions, with the simulation results without prediction, and with the results of the simulator using a less-intelligent predictor, namely the Neighbor Predictor.

We experiment with four bandwidth reservation strategies that install bandwidth reservations on behalf of all calls in progress, using the predictors to anticipate next handoff location and time:

- **Using all proportional probabilities *non-normalized*.** For some user, for instance, when the predictor outputs a vector:

$$< (A, 0.4), (B, 0.3), (C, 0.2) >,$$

we reserve 0.4Bandwidth Units (BU) at access point A, 0.3BU at access point B, and 0.2BU at access point C, where BU is the bandwidth required for a single voice call.
- **Using *normalized* proportional probabilities.** When, the predictor outputs a vector such as

$$< (A, 0.4), (B, 0.3), (C, 0.2) >,$$

we reserve $\frac{0.4}{0.4+0.3+0.2}$BU at access point A, $\frac{0.3}{0.4+0.3+0.2}$BU at access point B, and $\frac{0.2}{0.4+0.3+0.2}$BU at access point C, where BU is the bandwidth required for a single voice call.
- **Using top-3 normalized probabilities.** When, the predictor outputs a vector such as

$$< (A, 0.3), (B, 0.3), (C, 0.2), (D, 0.1), (E, 0.1) >,$$

we ignore all but the top three and normalize; that is, we reserve $\frac{0.3}{0.3+0.3+0.2}$BU at access point A, $\frac{0.3}{0.3+0.3+0.2}$BU at access point B, and $\frac{0.2}{0.3+0.3+0.2}$BU at access point C. We do not reserve any bandwidth at access point D or E. This method ignores the tiny probabilities, often numerous in the prediction vector of the static neighbor predictor or aggregate CDF predictor.
- **One BU reservation at top 3 APs.** When, the predictor returns a vector such as

$$< (A, 0.3), (B, 0.3), (C, 0.2), (D, 0.1), (E, 0.1) >,$$

we reserve 1BU each at access points A, B and C. As with the previous case, we ignore APs with small probabilities. This policy reserves up to 3BU, whereas the above policies reserve at most 1BU. The purpose is to compare with prior literature using this policy, and to avoid under-provisioning.

For the CDF-based predictors, all these strategies update the reservations whenever a user starts a call successfully, every T seconds thereafter (where T is a design parameter) until the call ends, and whenever the call is handed off to another AP. For the static neighbor predictor, these reservations are placed as soon as a user starts a call successfully and whenever the call is handed off to another AP; the reservations are never updated as the predictions do not change over time.

A given AP may hold reservations for many callers, and allows itself to be overbooked (that is, the total reservations may exceed the capacity of the AP). Any incoming handoff call will be accepted (not be dropped) whenever there is sufficient unused bandwidth regardless of how much bandwidth is reserved by this or any other caller; otherwise it is dropped. If the arriving caller had a reservation, its contribution to the reservation pool is also removed. A new call will be accepted whenever there is sufficient unused *and unreserved* capacity; otherwise it is blocked. Thus, handoff calls have priority over new calls, and reservations may reduce call drops but may increase call blocks.

We summarize our admission-control procedure with pseudocode. For each AP, let C denote the total capacity, U denote the bandwidth used by current calls, R denote the bandwidth reserved, and let R_u denote the reservation of user u at this AP, according to one of the above policies. For any AP, the reservation algorithm is as follows:

- Whenever reservations R_u are updated,
  $R = \sum_u R_u$;
- When a new call begins,
  if ($U + BU \leq C - R$) // unreserved bandwidth available
  then $U = U + BU$; // call allowed to begin
  else call is blocked;
- When user u with an active call handoffs to this AP,
  if ($U + BU \leq C$) // any bandwidth available
  then $U = U + BU$; // call allowed to continue
  else call is dropped;
  $R = R - R_u$;
- When a call ends or leaves the AP,
  $U = U - BU$;

B. Traffic Generation

As mentioned above, we generate a synthetic calling pattern for each of our users. The duration of a call is exponentially distributed with mean λ_d. The length of each user's inter-call time (between the end of one call and the beginning of the next call for the same user) is exponentially distributed with mean λ_i.

The Dartmouth network is not yet heavily loaded. To increase congestion, we set both the mean duration λ_d and the mean inter-call time λ_i to 15 minutes (900sec).

C. Training

The quality of predictions in the CDF predictors depends on whether, in the past, there have been enough observations at the APs where the user will be associated. To prevent such lack of observation, we train the predictor with the mobility data of the first month without making actual reservations, and start bandwidth reservations at all APs in the second month. Thus, when training is applied, we measure the application performance during the second month only. For the sake of completeness, we also provide the performance results with no predictor training.

D. Results

We ran our simulation using the following predictors:

- MarkovCDF O(2f) Individual predictor,
- MarkovCDF O(2f) Aggregate predictor,
- Static Neighbor predictor trained on the October 2003 data.

The two MarkovCDF predictors were run with and without training. All three predictors were run with the four reservation strategies described above.

We ran each case 10 times, each time with a different seed for the traffic-generation module. Unless otherwise mentioned, the AP capacity C=5BU and T is 300sec. We plot the relative improvement in the drop rate (Figure 8) and relative worsening of the block rate (Figure 9) as compared to a system without prediction-based bandwidth reservation. Note that these plots show the *means of the ratios*, with mean computed across the many seeds; that is, we compute for each predictor A

$$mean(\frac{\textit{DR} \text{ using no predictor}}{\textit{DR} \text{ using predictor A}}) \text{ and}$$

$$mean(\frac{\textit{BR} \text{ using predictor A}}{\textit{BR} \text{ using no predictor}}).$$

In the accompanying plots, Base *DR* is the *DR* using no predictor and Reserved *DR* is the *DR* after using a given predictor A. Similarly, Base *BR* is the *BR* using no predictor and Reserved *BR* is the *BR* after using predictor A.

The reason we use improvements and worsenings in the plots rather than the nominal values is that the DRs and the BRs vary greatly over the different seeds to the call generation module. Hence it makes more sense to summarize the *improvements* in the *DR* and the *worsenings* in the *BR*. To read the plots, if the value for the "Base *DR*/Reserved *DR*" ratio was 2, we say that the *DR* without prediction was double the *DR* with prediction. Similarly, if the value for the "Reserved *BR*/Base *BR*" ratio was 1.5, we say that the *BR* with prediction was 50% higher than the *BR* without prediction.

In Figure 8 and Figure 10 we observe that the Base *DR*/Reserved *DR* ratio lies between 1.4 and 22.4. From Table II we can see that the average *DR* with no prediction was 13.8%; prediction-based bandwidth reservation reduced *DR* to between 0.7% and 10%. We see that the individual predictors tended to perform better than the aggregate predictors in terms of *DR* and *BR*. This implies that an individual user was more likely to repeat her own patterns than to follow others' patterns at the same AP.

Intuitively, if the *DR* improves, the *BR* should worsen and vice versa. In an extreme case, the entire bandwidth would be reserved for handoff calls and all new calls would be blocked. It is expected that the choice of predictor depends on what *BR* one wishes to tolerate. Table II shows that on average the *BR* with no prediction was 11.7%. With reservation, the *BR* increases to between 14%-41%. The 40% *BR* was observed in the case of the

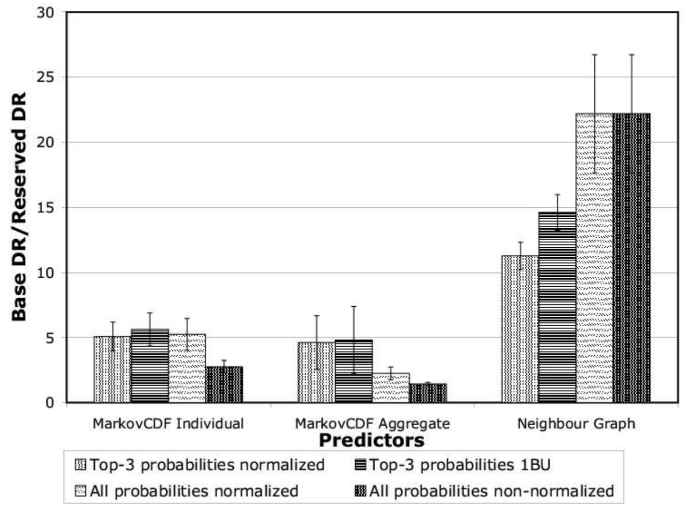

Fig. 8. The ratio (Base *DR*/Reserved *DR*) for all the Predictors with training. The error bars show the standard deviation σ among the various seeds of the simulation. Higher ratio is better.

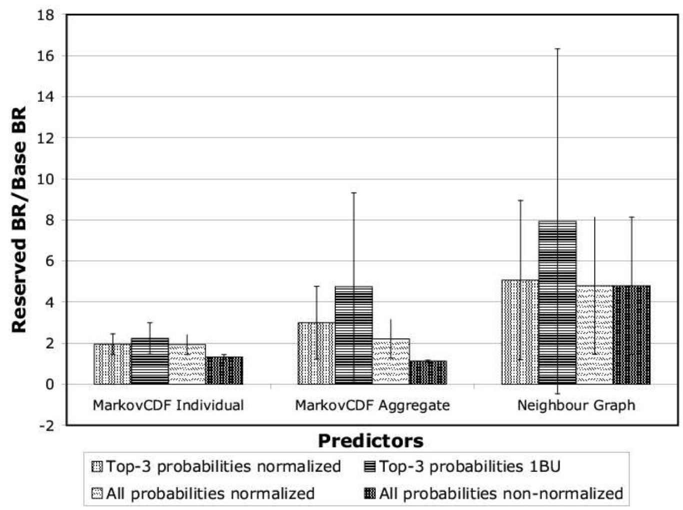

Fig. 9. The ratio (Reserved *BR*/Base *BR*) for all the Predictors with training. The error bars show the standard deviation σ among the various seeds of the simulation. Higher ratio is better.

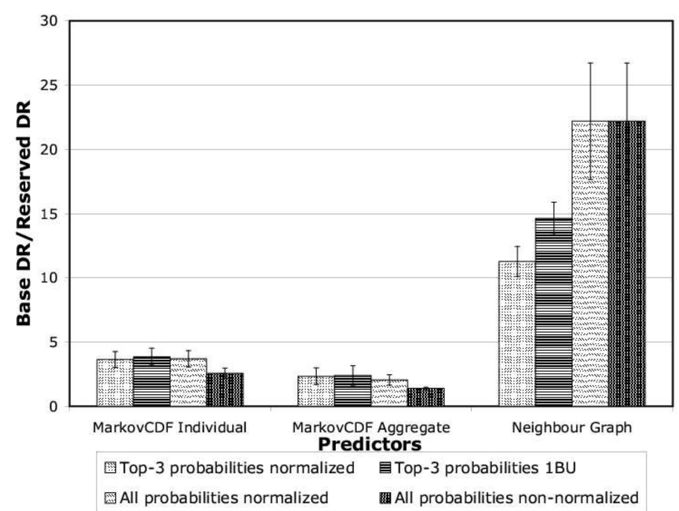

Fig. 10. The ratio (Base *DR*/Reserved *DR*) for all the Predictors, where MarkovCDF predictors had no training. The error bars show the standard deviation σ among the various seeds of the simulation. A higher ratio is better.

TABLE II

ABSOLUTE VALUES OF CALL DROP RATE (DR) AND BLOCK RATE (BR) WITH TRAINING

	Reservation Scheme	No Prediction		MarkovCDF Individual		MarkovCDF Aggregate		Neighbor Graph	
		Average†	σ‡	Average	σ	Average	σ	Average	σ
drop-rate (DR)	All normalized	13.80	7.58	2.98	2.23	7.01	4.73	0.69	0.51
	All non normalized	13.80	7.58	5.57	3.86	10.04	6.03	0.69	0.51
	Top-3 1 BU	13.80	7.58	2.74	2.01	4.36	3.53	0.96	0.58
	Top-3 normalized	13.80	7.58	3.03	2.23	4.28	3.48	1.24	0.74
block-rate (BR)	All normalized	11.74	10.35	18.72	14.1	18.84	12.98	33.23	19.31
	All non normalized	11.74	10.35	14.51	12.13	12.88	11.05	33.23	19.31
	Top-3 1 BU	11.74	10.35	20.24	14.50	26.81	13.51	40.61	17.66
	Top-3 normalized	11.74	10.35	18.85	14.14	22.56	14.15	33.48	18.94

†The values in the *average* column are means of the percentage rates over 10 seeds in the simulation.
‡The σ value is the standard deviation over these seeds.

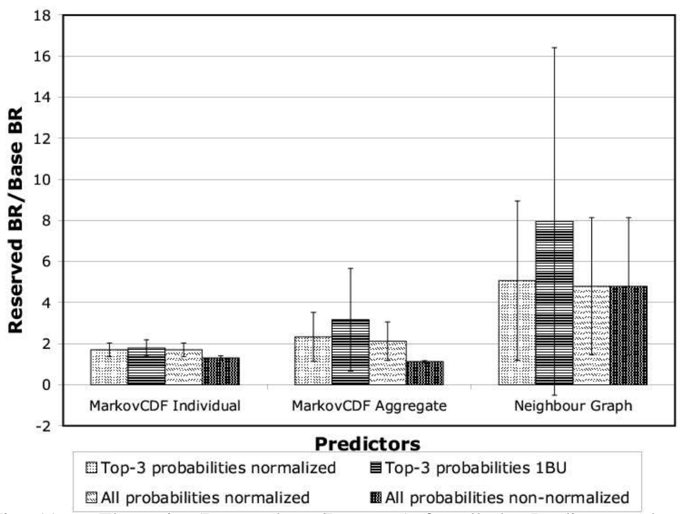

Fig. 11. The ratio (Reserved *BR*/Base *BR*) for all the Predictors, where MarkovCDF predictors had no training. The error bars show the standard deviation σ among the various seeds of the simulation. A higher ratio is better.

Neighbor predictor. It is clearly important to reserve bandwidth, but not too much. The ideal ratio may be achieved by the right choice of predictor and by tuning parameters such as T.

As we can see from the trained data set (Figure 8 and Figure 9), the MarkovCDF individual predictor came closer to achieving this balance than did the MarkovCDF aggregate predictor. Specifically, the MarkovCDF individual predictor improved the drop rate more than the MarkovCDF aggregate predictor, while at the same time it worsened the call-block rate to a lesser extent. We attribute the better performance exhibited by the individual predictor to the higher probabilities being returned by the MarkovCDF individual predictor for the more likely APs. Remember that the MarkovCDF aggregate predictor still tries to return predictions for individual users based on the hypothesis that most users move in a similar fashion at different locations. However, individual user behavior was not accurately predicted using aggregate history, probably because users were not as alike as expected. In the extreme, if the individual histories were independent from each other, then the aggregate history cannot statistically represent the individual moves that we want to predict.

Notice in Figure 9 that the standard deviation in the worsening of the BR was high, although the standard deviation for the improvement in DR was low. Hence, although there was an inverse relation between the DR and the BR on the average, the specific hand-off statistics and call patterns had significantly different effects in individual simulation runs, affecting BR more the DR. A stable DR is desirable, perhaps more so than a stable BR. Ideally, both DR and BR would be stable.

The "top-3 probabilities normalized" reservation scheme did not provide much benefit for the MarkovCDF individual predictor in terms of *BR*. The reason is that there are often not many more than 3 APs in the prediction vector of the MarkovCDF individual predictor anyway. So the limit of 3 that the "top-3 probabilities" scheme imposed is superfluous.

The MarkovCDF aggregate predictors benefited from the "top-3 probabilities normalized" scheme in terms of *DR*, because concentrating of reservations at fewer APs provided more resources for users on the network that have similar behavior. Due to these large reservations, however, there was more blocking and hence a higher *BR*.

The neighbor graph predictor performed worse using the "top-3 probabilities normalized" reservation scheme in both metrics. The drop rate did not improve as much as the "all probabilities" scheme because the "top-3" scheme made few reservations. The block rate worsens slightly more because of the concentrated reservations, similar to the MarkovCDF aggregate predictor.

In our simulation, the "top-3 probabilities $1BU$" scheme only provides slight benefit in *DR* over the others and worsens the *BR* considerably more. The reason is that it reserves $3BU$ per prediction rather than $1BU$, system wide. Hence any scheme with proportional probabilities is recommended.

For the two MarkovCDF predictors, the normalized version of the "all probabilities" reservation scheme improves *DR* more than the non-normalized version at the cost of higher *BR*. Clearly the reason is that there are larger reservations made on behalf of the user. The trade-off here is between *DR* and *BR*. The Neighbor predictor is unaffected because the probabilities it returns are normalized to begin with.

While comparing the simulations with the trained predictors against the simulations with the untrained predictors we recall that the neighbor predictor is trained with the first month's data regardless, but it did not learn during the second month like the MarkovCDF predictors do. As expected, the training in the MarkovCDF predictors generally benefits the *DR* while worsening the *BR* to a lesser extent.

These results show that even with moderate prediction accuracy, we can improve the *DR*, while not worsening the *BR* as much. We see that the simplistic neighbor graph predictor will block considerably more calls than is acceptable in most systems, because it cannot predict time at all. Using the predictors that we have presented helps us to be more discriminating in our reservations. Furthermore, the choice of the reservation schemes with the predictors is important.

VIII. SUMMARY

In this paper, we focus on quantifying the quality of mobile handoff predictions in a real large-scale WLAN environment. We experiment with a variety of mobility predictors, all of which predict the time of a mobile user's next handoff event, and some of which jointly predict the destination as well. We select predictors such that their success would depend on certain dependencies across space and time. We further investigate whether group mobility or individual mobility models better represent the real situation. After evaluating these predictors with a suite of performance metrics, we find that no single predictor feature performed uniformly well, i.e., the prediction quality varies widely from user to user and from access point to access point. In the over- and under-provisioning results, for example, we find the distributions to be highly skewed.

Our results show that predicting the precise time of the handoff with a granularity in the order of seconds and minutes is in general not possible by using only the association history of the mobile devices. These results drive us to explore predictors that can return a range of values, can express confidence in their prediction (a probability), and can express inequalities (the probability that the visit will last at least t seconds, rather than predicting departure after precisely t seconds). We observe that predictions for individual events still tend to be highly skewed even for such *soft* prediction outputs. This motivates us to look into applications that could manage such skewness and benefit from soft values. Thus, we look at a popular application, i.e., VoIP over WLAN, that can use per-user predictions to reserve resources for a group of users (in our case, the handoff users). Indeed, our results demonstrate that when coupled with intelligent predictors, VoIP performance improved significantly over the cases when only simple predictors are allowed or when no reservations are made at all.

Note that our main concern in this paper is on quantifying how campus WLAN mobility data was amenable to useful predictions. Therefore, we do not pay any particular attention to predictor complexities and efficient system implementation details, which can be tackled once the significant advantages of mobility predictions can be shown. This challenge will be part of our future research.

ACKNOWLEDGEMENT

This research is a project of the Center for Mobile Computing and the Institute for Security Technology Studies at Dartmouth College. It was supported by DoCoMo Labs USA, the CRAWDAD archive at Dartmouth College (funded by NSF CRI Award 0454062), NSF Infrastructure Award EIA-9802068, and by Grant number 2005-DD-BX-1091 awarded by the Bureau of Justice Assistance. Points of view or opinions in this document are those of the authors and do not represent the official position or policies of any sponsor.

REFERENCES

[1] Songwu Lu and Vaduvur Bharghavan, "Adaptive resource management algorithms for indoor mobile computing environments," in *Proc. ACM SIGCOMM '96*, August 1996, pp. 231–242.

[2] David A. Levine, Ian F. Akyildiz, and Mahmoud Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 1, pp. 1–12, Feb. 1997.

[3] Anup K. Talukdar, B.R. Badrinath, and Arup Acharya, "On accommodating mobile hosts in an integrated services packet network," in *Proc. IEEE INFOCOM 1997*, Apr. 1997, pp. 1046–1053.

[4] Tong Liu, Paramvir Bahl, and Imrich Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks," *IEEE JSAC*, vol. 16, no. 6, pp. 922–936, August 1998.

[5] Carlos Oliveira, Jaime Bae Kim, and Tatsuya Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE JSAC*, vol. 16, no. 6, pp. 858–874, August 1998.

[6] Sunghyun Choi and Kang G. Shin, "Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks," in *Proc. ACM SIGCOMM '98*, August 1998, pp. 155–166.

[7] Xuemin Shen, Jon W. Mark, and Jun Ye, "User mobility profile prediction: An adaptive fuzzy interference approach," *Wireless Networks*, vol. 6, no. 5, pp. 363–374, 2000.

[8] Ming-Hsing Chiu and Mostafa A. Bassiouni, "Predictive schemes for handoff prioritization in cellular networks based on mobile positioning," *IEEE JSAC*, vol. 18, no. 3, pp. 510–522, March 2000.

[9] AbdulRahman Aljadhai and Taieb F. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE JSAC*, vol. 19, no. 10, pp. 1915–1930, October 2001.

[10] Wee-Seng Soh and Hyong S. Kim, "Dynamic guard bandwidth scheme for wireless broadband networks," in *Proc. IEEE INFOCOM 2001*, April 2001.

[11] Fei Yu and Victor Leung, "Mobility-based predictive admission control and bandwidth reservation in wireless cellular networks," *Computer Networks*, vol. 38, no. 5, pp. 577 – 589, April 2002.

[12] Wee-Seng Soh and Hyong S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility prediction," in *Proc. IEEE INFOCOM 2004*, Hong Kong, March 2004.

[13] Arunesh Mishra, Min ho Shin, and William A. Arbaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *Proc. IEEE INFOCOM 2004*, March 2004, pp. 351–361.

[14] Zainab R. Zaidi and Brain L. Mark, "Real-time mobility tracking algorithms for cellular networks based on Kalman filtering," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 195– 208, March-April 2005.

[15] Libo Song, David Kotz, Ravi Jain, and Xiaoning He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. IEEE INFOCOM 2004*, March 2004.

[16] Songwu Lu and Vaduvur Bharghavan, "Adaptive resource management algorithms for indoor mobile computing environments," in *Proc. ACM SIGCOMM '96*, August 1996, pp. 231–242.

[17] Fang Feng and Douglas S. Reeves, "Explicit proactive handoff with motion prediction for Mobile IP," in *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, March 2004, vol. 2, pp. 855–860.

[18] David Kotz and Kobby Essien, "Analysis of a campus-wide wireless network," *Wireless Networks*, vol. 11, pp. 115–133, 2005.

[19] Tristan Henderson, David Kotz, and Ilya Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc. ACM MobiCom 2004*, September 2004, pp. 187–201.

[20] Christine Cheng, Ravi Jain, and Eric van den Berg, "Location prediction algorithms for mobile wireless systems," in *Handbook of Wireless Internet*. CRC Press, 2003.

[21] Zainab R. Zaidi and Brain L. Mark, "Mobility estimation for wireless networks based on an autoregressive model," in *Proc. IEEE Globecom '04*, November 2004.

[22] Pubudu N. Pathirana, Andrey V. Savkin, and Sanjay Jha, "Mobility modelling and trajectory prediction for cellular networks with mobile base stations," in *Proc. ACM MobiHoc'03*, 2003, pp. 213–221.

[23] Amiya Bhattacharya and Sajal K. Das, "LeZi-Update: An information-theoretic approach to track mobile users in PCS networks," *ACM/Kluwer Wireless Networks*, vol. 8, no. 2-3, pp. 121–135, March–May 2002.

[24] Philippe Jacquet, Wojciech Szpankowski, and Izydor Apostol, "An universal predictor based on pattern matching, preliminary results," in *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities*, chapter 7, pp. 75–85. Birkhauser, 2000.

[25] Jean-Marc François and Guy Leduc, "Mobility prediction's influence on QoS in wireless networks: A study on a call admission algorithm," in *Proc. IEEE WiOpt 2005*, April 2005, pp. 238 – 247.