Model T: An Empirical Model for User Registration Patterns in a Campus Wireless LAN

Ravi Jain
jain@docomolabs-usa.com

Dan Lelescu
lelescu@docomolabs-usa.com

Mahadevan Balakrishnan
mahadevan@docomolabs-usa.com

DoCoMo Communications Labs USA
181 Metro Drive, Suite 300, San Jose, CA. 95110

ABSTRACT

We derive an empirical model for spatial registration patterns of mobile users as they move within a campus wireless local area network (WLAN) environment and register at different access points. Such a model can be very useful in a variety of simulation studies of the performance of mobile wireless systems, such as that of resource management and mobility management protocols. We base the model on extensive experimental data from a campus WiFi LAN installation, representing traces from about 6000 users over a period of about 2 years. We divide the empirical data available to us into training and test data sets, develop the model based on the training set, and evaluate it against the test set.

The model shows that user registration patterns exhibit a distinct hierarchy, and that WLAN access points (APs) can be clustered based on registration patterns. Cluster size distributions are highly skewed, as are intra-cluster transition probabilities and trace lengths, which can all be modeled well by the heavy-tailed Weibull distribution. The fraction of popular APs in a cluster, as a function of cluster size, can be modeled by exponential distributions. There is general similarity across hierarchies, in that inter-cluster registration patterns tend to have the same characteristics and distributions as intra-cluster patterns.

We generate synthetic traces for intra-cluster transitions, inter-cluster transitions, and complete traces, and compare them against the corresponding traces from the test set. We define a set of metrics that evaluate how well the model captures the empirical features it is trying to represent. We find that the synthetic traces agree very well with the test set in terms of the metrics. We also compare the model to a simple modified random waypoint model as a baseline, and show the latter is not at all representative of the real data.

The user of the model has the opportunity to use it as is, or can modify model parameters, such as the degree of randomness in registration patterns. We close with a brief discussion of further work to refine and extend the model.

MobiCom'05, August 28–September 2, 2005, Cologne, Germany.

Categories and Subject Descriptors: C.2.1 Wireless communication, I.6.5 Modeling methodologies.

General Terms: Design, performance, measurement.

Keywords: Mobility models, registration models, wireless LAN, simulations.

1. INTRODUCTION

The design of protocols and algorithms for mobile wireless networks is often evaluated by simulation. For example, to help evaluate a mobility management protocol such as Mobile IP, a protocol designer might set up a test scenario of a number of wireless access points; simulate the way users move between the access points, register at the access points, and generate traffic; and infer the signaling overhead that results. One of the critical elements of such a simulation would be the registration patterns of users at the different access points.

One way to generate such registration patterns is to assume geographical locations for the access points, position users at points in the geographical space, and then simulate the actual movement of users with a user mobility model as input. Several mobility models have been developed, such as random walk, random waypoint [1] and the obstacle model [2].

In the scenario described above, where the object is to understand the behavior of a mobility management protocol, the user mobility model is used to generate a registration pattern for the users; the actual mobility of the users in geographical space is not very important. Clearly a user mobility model can be more general and more useful than a model of user registrations, but for many applications the latter may suffice. While we would like to be able to develop a user mobility model based on empirical data, we currently do not have data available to us to do so; *this paper thus focuses on developing an empirical model of user registration patterns only, and not of geographical mobility.*

Developing a model of user registrations turns out to be non-trivial. Our aim is to develop a model that can be used in the following way. The modeler (e.g., a protocol designer) would specify a relatively small number of parameters, and the model would probabilistically generate registration traces consistent with those parameters. In particular the modeler would specify the number of users to be modeled, and some aggregate characteristics of their registration patterns.

This paper describes the design of such an empirical model. To develop our model we use a large set of user mobility

traces collected from the Dartmouth College campus WiFi network, which recorded the time and identity of over 500 wireless cells visited by more than 6000 users over a period of over 2 years [3]. This data has some limitations for our purposes, but is the best available to us at present.

One of the first, important, observations we make from the experimental data is that user registration patterns are highly skewed in space distributions, and do not resemble a random waypoint or similar random pattern at all. Another is that the spatial distribution of user registration patterns is hierarchical, i.e., there are clusters of access points among which a significant majority of the changes in user registrations occur, and relatively few transitions from one cluster to another.

These observations are captured in a quantitative model, and can be useful for protocol designers, for example, as follows. Suppose the designer is evaluating a new protocol. A model that captures the locality of registrations may favor certain protocol designs (e.g., those that rely on caching of user registrations). This can also be a guide or influencing factor for protocol design itself. The existence of clusters and their distribution might motivate the designer to consider protocols, for example, that rely on electing local cluster heads for signaling and control.

Our contributions in this paper, based on the experimental data set we use, are as follows: (1) we develop an extensive statistical characterization of registration patterns; (2) we show that simple models such as random waypoint cannot be used to realistically reflect user registration behavior; (3) we divide the empirical data into a test and training set, and develop a user registration model for inter-cluster and intra-cluster patterns based on the training set; (4) we show that the registration traces synthesized from model derived from the training set is in good agreement with the test set. We also briefly illustrate the flexibility of the model in allowing the model user to modify its parameters in intuitive ways. We also discuss ways in which the model can be refined, albeit at the expense of increased complexity.

The outline of the rest of this paper is as follows. In section 2 we describe the background for this work, include related work, and the approach we take. In section 3 we summarize how the experimental data was collected, and its limitations for our purpose. In section 4 we describe the empirical model we have developed, which we call *Model T*. In section 5 we describe its evaluation against the data; we also compare it against a modified random waypoint model as a baseline. Note that *we present Model T only in terms of spatial registration patterns and not temporal patterns*, i.e., the pattern of registration changes from one AP to another, and not the times at which they occur or the residence time durations at APs. In section 6 we discuss different variations of Model T that are possible. We also discuss how Model T can be trivially augmented with an independent time model, of the type used in models such as random walk and random waypoint; we point out that developing a joint empirical spatial-temporal model is a challenging problem and outside the scope of this paper. A discussion of subtleties of the model is presented in section 7. Finally in section 8 we end with some conclusions and brief remarks about further work.

Terminology. We will use the following terminology. When a user disconnects from the network (e.g., powers off a laptop), this is called the OFF state. A *transition* is a registration by the user's device at one access point (AP) followed by a registration at a different AP. Sometimes a transition may be referred to informally as a *move,* although we stress again that we are not developing a user mobility model.

2. BACKGROUND AND APPROACH

2.1 Related work

To our knowledge there is no previous work that focuses explicitly on developing a mobile user registration model, other than the work in [4] which we discuss at the end of this subsection.

Previous work on user mobility models is the closest to our work. It typically focuses on modeling how the user moves in a geographical space, which can then be fed as an input to a simulation of, for example, a network protocol. The geographical movement of users can then be mapped to network events (e.g., registrations or handoffs) that are of interest. A wide variety of theoretical models have been developed for this purpose; see Camp [1] for a survey.

The random walk model is especially popular in network protocol simulations. Although there are numerous variations, the basic idea is that the user selects a random direction from [0, 2π] and a random speed from a specifiable range [v_{min}, v_{max}], and moves with those parameters, stopping either until a specified time or distance has elapsed. Another popular model is the random waypoint model. It also has several variations, but it is similar to random walk except that instead of picking a random direction, a random destination is chosen, and the user may then pause for some specifiable time or interval range before repeating the process.

The advantages of theoretical models are that they are simple to understand and implement, and can even lend themselves to algebraic analysis in some situations. More importantly, they are *scalable*, in the sense that they allow the model developer to choose the number of cells and their layouts, as well as the number of users. However, there are notable disadvantages. Both the random walk and random waypoint, for example, have been shown to exhibit very unrealistic or undesirable behaviors, such as sudden stops and turns or model initialization problems [2], [5].

Recent work on user mobility models starts with a theoretical model and then tries to make it more realistic by incorporating random waypoints, random destinations, or modeling physical obstacles [2]. This approach is certainly interesting. However, instead of trying to make a theoretical model of registration patterns (e.g., starting with a theoretical user mobility model and deriving a registration model) we start with empirical data and try to develop a realistic model of registrations.

Recent work [6] has started taking steps along this approach. However, we believe our work is more focused on directly developing a user registration model, and is, to our knowledge, the only comprehensive attempt to develop a model using very extensive experimental data from campus WLAN traces.

One important piece of related work is the effort that has gone into collecting and carrying out basic analysis of traces of user registrations, in particular the work of Kotz and colleagues [3]. Our modeling rests on this previous work and is indebted to it. However, it is important to point out that this previous work does not attempt to build a quantitative model of user registrations in the way we do

(indeed, a large part of previous such studies focuses on the characteristics of the data traffic). The analysis in this paper is significantly deeper, including deriving statistical models of spatial patterns, the use of statistical clustering techniques to understand hierarchy, definition of metrics to compare the model with empirical data, and so on; it is thus quantitatively more detailed. In addition the previous work cannot directly be used to synthesize registration traces.

Some interesting works such as [7] and [8] have been done in the past which try to capture some statistical details associated with a WLAN network and characterize user mobility. These have hinted at the presence of popular locations where the users tend to visit frequently. In our paper we have characterized these as popular APs. Also [8] mentioned that the characteristics of the user behavior in various locations are different. In comparison to this, we explicitly modeled user mobility in clusters discovered from the real data. Though [7] and [8] have analyzed the user mobility and other characteristics of the network (e.g., throughput), they do not model them and simply present the statistics associated with the network. Further, we extracted various statistical features such as clustering and popularity of clusters, and modeled these features based on applying statistical fitting to the extracted information with an eye towards trace synthesis. The experimental data we used is more extensive in terms of both the number of users and duration of observation.

Finally, our previous work [4] has the same motivation and approach, and uses the same data set. However, that work primarily offers a quantitative characterization of the time features of user registration patterns. These are the mean time duration that a user registers at any given AP (the residence time), the mean time that a user is OFF (the OFF time), and the mean number of OFF states per day. It does make some initial observations about hierarchy and locality in spatial registration patterns, but these are not quantified or validated in any way. In contrast the present paper develops a fairly complete model of the spatial registration models, instantiates its quantitative parameters for the given data set, defines metrics to evaluate the model, evaluates the metrics, and illustrates how the model parameters can be modified.

2.2 Approach

One could envision registration models at different levels of abstraction. The most elementary "model" is simply the user's registration traces themselves, which can be input to a trace-driven simulation program. A more abstract (but still basic) model, at least for modelling spatial characteristics of mobility, is to construct a Markov transition probability matrix $M(i, j)$, which extracts from the data the probability that a user at AP i will transition to AP j. (This approach can be easily generalized to higher-order Markov models). However, such basic models do not allow the model user to easily vary the number of users or the parameters of the model or both.

The basic approach we use to develop the registration model is an iterative process with the following steps: (1) mine and explore the experimental data to understand it; (2) divide the data into a training set and a test set; (3) develop a model as a set of equations that characterize the salient features of the training set; (4) generate synthetic traces from the model; (5) compare the synthetic traces with the test set, and possibly with other models; (6) repeat starting at step (1) with improved understanding. In this paper we present Model T, the result of our iterations through this process.

Any model derived from experimental data is limited by that data set. In our case, clearly Model T will be limited to campus WLAN or similar environments, and is unlikely to model, say, cars on highways. It is also likely to be influenced by the specifics of the Dartmouth campus user population, AP layout, building layout and possibly even environmental factors such as climate. Our claim is not that Model T completely replaces other approaches such as using random waypoint models, obstacle models or trace-driven simulations. Nonetheless, judiciously used, it can offer protocol designers and networking researchers a significant additional tool for understanding and evaluating their designs.

3. EMPIRICAL DATA

The experimental data set we use was collected at Dartmouth College [3], providing a nearly continuous, two-year record from April 2001 through March 2003. The data covers 586 APs in 161 buildings spread over about 200 acres, with registration patterns from 6202 users. The APs provide 11 Mbps coverage to the entire campus during this period; since the campus is compact the interior APs tend to cover most outdoor spaces.

The APs transmit a "syslog" message every time a WiFi client card associated, re-associated, or disassociated with the AP; the message contains the unique MAC address of the client card. A user's trace refers to the record of a single user's registration-related events, including the OFF state, which represents the user's departure from the network (by turning off the device or leaving the range of all APs).

Although a given card might be used in multiple devices or a device used by multiple people, in this paper we think of the wireless card as a "network user" and thus the term "user" refers to a wireless card. The locations in a trace are the names of the access points with which a user is associated and do not necessarily correspond to the precise geographical locations of humans. Similarly, changes in location, i.e., moves, do not necessarily correspond to physical movements but changes in registrations. A limitation of the data is that the OFF state is not always accurately captured. Further details of the data and collection process are described in [3, 9].

Another issue is that we do not have available the physical location coordinates of some of the APs (about 100 or so out of 586). These AP location inaccuracies come into play when we investigate the distribution of transition probabilities with inter-AP distance, as described later.

4. REGISTRATION MODEL

4.1 Gross statistics

We first consider whether flow is conserved at individual APs, i.e., whether the total number of incoming transitions over all users approximately equals the total number of outgoing transitions. This acts as a sanity check for the data itself as well as for our post-processing steps. We find that in fact flow conservation holds, and the maximum deviation observed is at most 3%, with the exception of building Bradley, where the number of outgoing transitions exceeded

the incoming by about 20%. This turns out to be the campus computer store, where laptops and wireless cards are first activated.

We then considered whether flow between APs was symmetric, i.e., for any pair of APs, if the number of transitions from one to the other approximately equals the number in the reverse direction. Here we found that in fact significant asymmetry can exist, with up to 23% more flow in one direction that the other. Thus a transition probability matrix for all the APs in the campus need not be symmetric along the main diagonal.

4.2 Locality and hierarchy

Previous work [9] has shown that the trace length for different users shows very wide variability, with a median of 494 and a maximum of 188,479 APs visited. However observe that, on average, users visit only a few *distinct* APs per day that they are active; see Fig. 1.

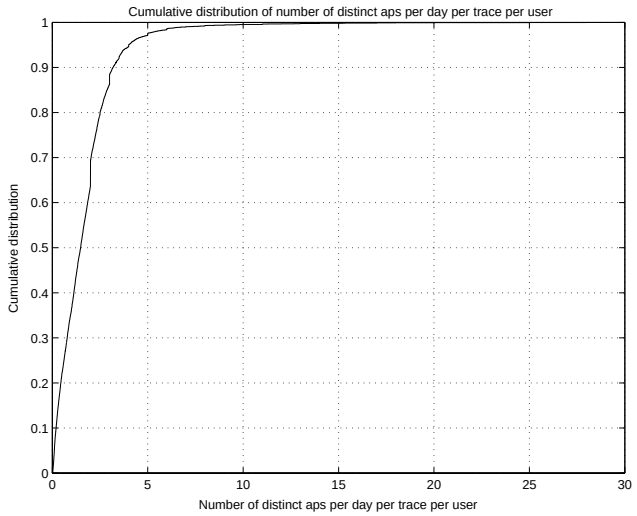

Figure 1: Number of distinct APs per day per trace per user.

Note that this by itself suggests but does not definitely imply that there is a great deal of spatial locality in registrations. To investigate further we consider the registration pattern of all users over all the APs for the entire length of each trace.

We first build a *transition graph*, where each vertex represents an AP and an edge connects two vertices iff there is at least one transition between the corresponding APs. Each edge (u, v) is labelled with the probability that a user at AP u will make a transition to AP v ; for this purpose we take into account the OFF state.

We then apply the graph clustering tool MCL (Markov clustering) [10]. Applying MCL to the campus-wide transition graph reveals a set of clusters, giving direct evidence of locality and hierarchy. The MCL tool generated 86 clusters ranging in size from 1 to 25 APs in a cluster. The CDF of cluster sizes is shown in Fig. 2. The fit to the empirical CDF distribution shown in Fig. 2 is given by a Weibull CDF whose equation is:

$$F(x) = 1 - e^{-(\frac{x}{a})^b}. \quad (1)$$

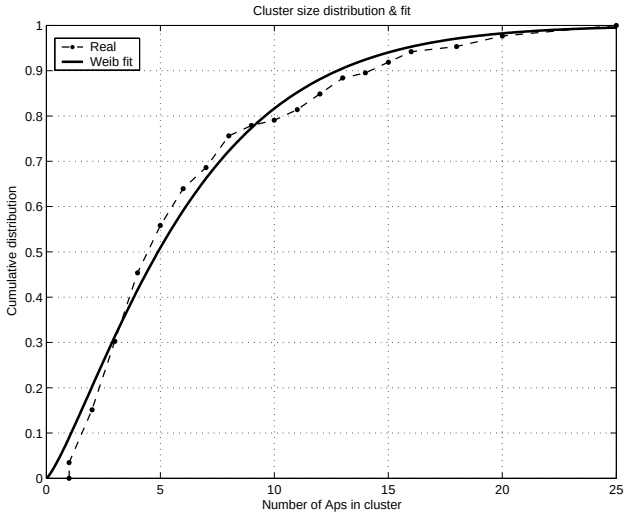

Figure 2: Empirical CDF and fit for cluster size distribution.

The parameters of this fit are $a = 6.561$, $b = 1.253$. The goodness of fit (GOF) measures are $rmse = 0.03892$ and $R^2 = 0.9858$.

We observe that while clustering may seem like an intuitively appealing idea when considering registration patterns in the abstract, to our knowledge this study is the first analysis that demonstrates clustering for empirical data from pedestrian wireless users. Further, the analysis shows that most cluster sizes are small; the median is 5 APs per cluster.

We examined the clusters discovered from the experimental data, and found that they do not correspond to campus structures such as buildings. On reflection this is not surprising for a compact campus, since APs in neighboring buildings may frequently serve users at the edge of the building.

Each AP is assigned to exactly one cluster. We consider a transition in a user registration trace to be an *intra-cluster transition* if it occurs between two APs of the same cluster. For this purpose the OFF state is ignored, i.e., if a trace contains a sequence $< a, OFF, b >$ where a and b are APs in the same cluster, this is regarded as a single intra-cluster transition; if a and b are in different clusters, it is regarded as a single inter-cluster transition. The percentage of intra-cluster moves in the entire set of traces is found to be 74.47%. Since a large majority of transitions are intra-cluster, we first focus on analyzing these transitions in the following.

In the remainder of this section, for developing the model, we divide the empirical data into a *training* and a *test* set, and use only the training set. In the following section, we compare the model derived in this section with the test set. The division of the empirical data is done by randomly selecting half the users represented in the data and assigning their traces to the training set, while the other half are assigned to the test set.

4.3 Intra-cluster modeling

We derive intra-cluster transition probability matrices from the data. For a cluster C, let the intra-cluster transition matrix be the $|C| \times |C|$ matrix N where $N(u, v)$ is the number

of intra-cluster transitions from AP u to AP v in C. The intra-cluster transition probability matrix M for C is such that

$$M(u,v) = \frac{N(u,v)}{\sum_i N(u,i)}. \quad (2)$$

Modeling the transitions within a cluster amounts to trying to isolate the salient features of how transition probabilities are distributed for intra-cluster moves. In particular, an interesting question is whether all transitions are roughly equally likely, which would be the case for a random walk model, or whether there exists some other law governing this process. Also, how does this process vary with cluster size? In order to answer these questions we mine the registration traffic data and attempt to identify the important features that might be used to characterize this process.

4.3.1 Transition probability distributions

We examine the distribution of the transition probabilities between pairs of APs within clusters, and find they are highly skewed, i.e., a relatively small number of AP pairs account for a large proportion of the transitions that occur within a cluster. As an example, the empirical CDF curve (labelled "Real") in Fig. 3 shows the distribution of transition probabilities in M for all 3 clusters of size 12 APs.

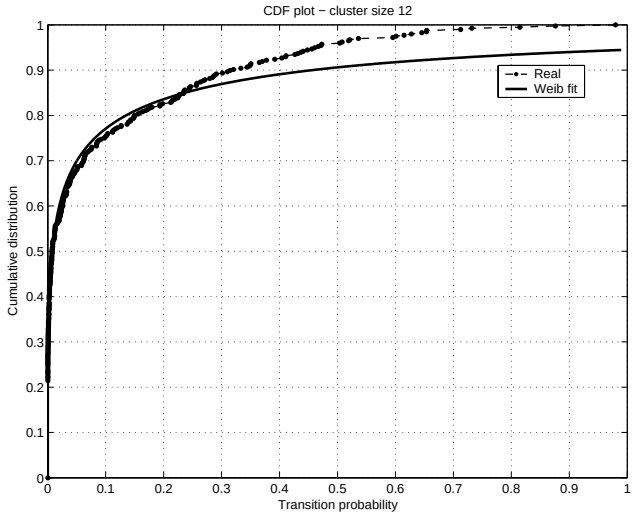

Figure 3: Empirical CDF and fit for intra-cluster transition probabilities for clusters of size 12.

We observe that the curves for the empirical CDFs of the transition probabilities generated for different cluster sizes become more skewed as the cluster size increases. This process is evident in the sample CDFs provided in Figs. 3 - 4. For small cluster sizes the intra-AP transition probability mass is more equally distributed among the APs in the cluster, whereas as the number of APs in the cluster increases this distribution is more uneven. For example in Fig. 4 we see that about 90% of the transition probability values have a probability that is less than or equal to 0.1. This fact indicates the presence of very popular APs toward which most of the movements are directed to.

So far we have been analyzing the empirical data only. Now we consider how we could use this information, for example in Figs. 3 - 4, to generate synthetic traces of user

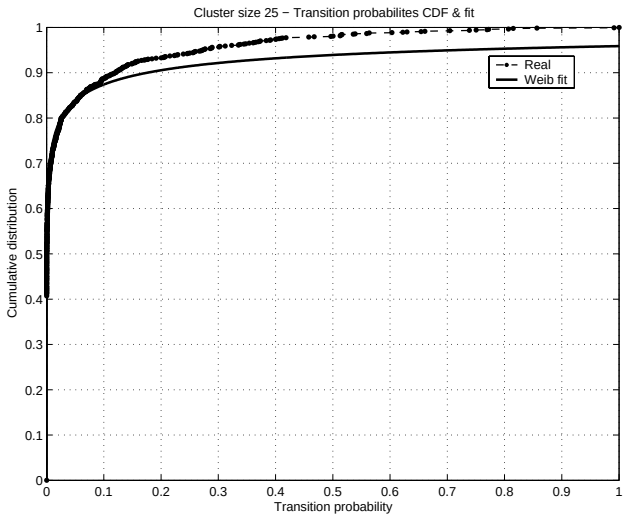

Figure 4: Empirical CDF and fit for intra-cluster transition probabilities for clusters of size 25.

registration patterns. We see that we must obtain a more abstract representation of this information. The easiest way to do this is simply to fit parameterized statistical distributions to the real data transition probability CDFs. Then during the trace synthesis we can generate traces by drawing transition probabilities from the statistical distributions. We create, for each cluster C the transition probability matrix M. We use the curve fit to the intra-cluster transition probability CDF, and populate the entries of the matrix, row by row, by drawing values from the CDF. The matrix will then reflect the skewed nature of transition probabilities; for each row, a few columns will account for most of the outgoing transitions from that AP.

The curve fits to empirical transition probability CDFs were obtained using the Curve fitting toolbox in MATLAB under criteria for goodness of fit such as *rmse*, R^2. Different distributions including Gaussian, log normal, Exponential, Gamma were tested and the best fit was obtained with a Weibull distribution which has a CDF parametrization given in Eq. (1). The Weibull distribution has its scale and shape controlled by the two a and b parameters. This distribution is general in that its shape and scale can vary, and by proper choice of parameters it can reduce to another common distribution (i.e., exponential).

The Weibull distribution fits well the data for all the cluster sizes, with a R^2 value of at least 0.88 and a *rmse* value of at most 0.082. Typically R^2 values are over 0.95 (as shown in Table 1). Recall that the Weibull is a heavy-tailed (but not a power-law) distribution. The parameters a and b of the distribution vary with the cluster size. This dependency is illustrated in Table 1, which shows the cluster size and the corresponding parameters a, b to be used for generating the required Weibull CDF.

A technicality worth noting is that a Weibull CDF fit to the empirical CDF of the real data never reaches 1 (as seen for example in Figs. 3 - 4). This is due to the fact that the equation of the Weibull CDF evaluated at $x = 1.0$ does not equal 1, and a small residue $\epsilon > 0$ remains. To generate proper transition probabilities in the $[0, 1]$ interval, we simply choose a CDF value z at random (uniformly)

Table 1: Weibull parameters for fitting transition probability CDFs for different cluster sizes

Cluster size	a-param	b-param	rmse	R^2
3	0.5219	0.6975	0.08229	0.88
4	0.3476	0.8074	0.03422	0.98645
5	0.1441	0.3833	0.04781	0.95926
6	0.1268	0.4373	0.02844	0.98827
7	0.0556	0.2846	0.04979	0.9468
8	0.1023	0.5373	0.0282	0.98851
9	0.08337	0.4977	0.03056	0.98613
10	0.06446	0.518	0.02882	0.98825
11	0.03375	0.3337	0.02232	0.9919
12	0.03668	0.3915	0.02101	0.9919
13	0.02331	0.3012	0.03408	0.97389
14	0.01602	0.2945	0.02525	0.98768
15	0.01509	0.2873	0.02892	0.98242
16	0.03024	0.4796	0.01383	0.99723
18	0.01106	0.2938	0.0252	0.98621
20	0.021	0.4568	0.01502	0.99648
25	0.005246	0.2561	0.02377	0.98437

between $[0, 1-\epsilon]$. Then the transition probability value is determined by selecting the value on the x-axis of the CDF that corresponds to z.

Thus, in principle, Model T will have one equation of the form in Eq. (1) for each cluster size; we discuss this point again next.

4.3.2 Reducing the number of parameters

In order to further develop the modeling of the intra-cluster transitions, we note that in an initial model, for each cluster size we could have a Weibull distribution fit to the transition probability data, with different associated parameters. This would imply a per-cluster parametrization of the corresponding transition probability CDFs. Rather than maintain multiple CDF models corresponding to each cluster size (which is possible), we model the variability in the Weibull a and b parameters with respect to cluster size. This is done in order to provide a higher level of abstraction and reduce the number of parameters of the model. Then, a comparison between the fitted Weibull CDFs for each cluster size (as previously described), and the corresponding *modeled* Weibull CDFs generated for each cluster size (based on obtaining a and b parameters from the parametric fits described below) enables an assessment of the modeling performance.

From Table 1, we observe that as the cluster size increases the a parameter decreases, indicating that the transition probabilities become more and more skewed. The b parameter remains less than 1, thus preserving the shape attributes of the CDF and has a smaller influence on the curve fit. The dash lines in Figs. 5, 6 illustrate the parameters of the Weibull CDFs fitted to each real transition probability CDF for each cluster size (see examples in Figs. 3 - 4 and Table 1). Different fits were tried for the a and b parameter values of the Weibull CDF. The exponential fit provided the best goodness of fit measures (R^2 and $rmse$). A fit to these curves using Eq. (3) is illustrated by the exponential curves shown in Figs. 5, 6.

$$y = p_1 e^{-p_2 x} + p_3. \quad (3)$$

The parameters of the fit to the variation of parameter a with cluster size are given by $p_1 = 2.55$, $p_2 = 0.5437$, and $p_3 = 0.02643$. For this fit, the $rmse = 0.0259$ and $R^2 = 0.9688$. Similarly, the fit to the variation of the parameter b with cluster size is given by $p_1 = 1.211$, $p_2 = 0.3802$, and $p_3 = 0.3552$. For this fit, the $rmse = 0.115$ and $R^2 = 0.5112$.

Therefore, using this more parsimonious description, we can generate CDFs of transition probabilities for a cluster of a given size, by determining their Weibull parameters according to the exponential law governing the dependency of these parameters a and b on the cluster size. Despite

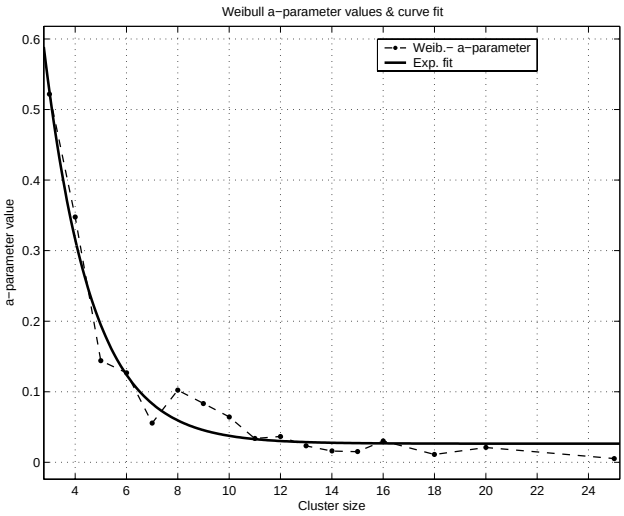

Figure 5: Exponential fit to Weibull a-parameter variation.

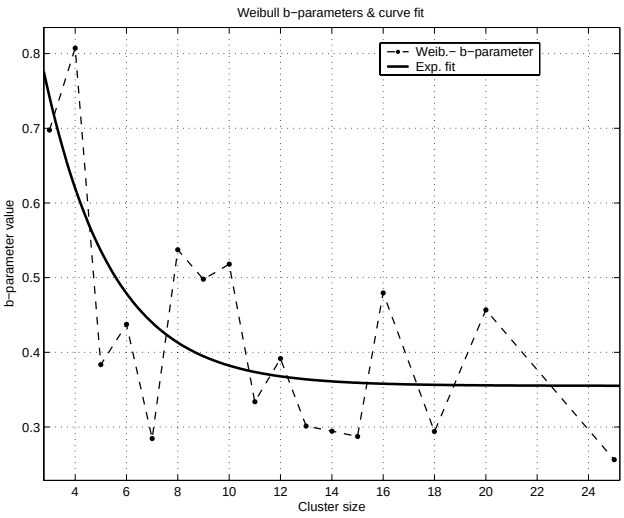

Figure 6: Exponential fit to Weibull b-parameter variation.

the poorer parametric description for the parameter b, the fit between a Weibull CDF modeling empirical data and the corresponding Weibull CDF synthesized using Eq. (3), for a given cluster size, is very good as shown by the goodness of fit measures in Table 2.

Table 2: Goodness-of-fit measures between empirical-fit CDFs and the modeled CDFs

Cluster size	rmse	R^2	Cluster size	rmse	R^2
3	0.00867	0.9996	12	0.00355	0.9953
4	0.05284	0.9872	13	0.01740	0.9846
5	0.05047	0.9549	14	0.02261	0.9954
6	0.05143	0.9902	15	0.02816	0.9724
7	0.06240	0.9360	16	0.04895	0.9967
8	0.06083	0.9630	18	0.03348	0.9600
9	0.04453	0.9780	20	0.06453	0.9823
10	0.06554	0.9153	25	0.07380	0.9228
11	0.01321	0.9997			

4.3.3 Number of popular APs

The intra-cluster transition probability CDF equations, as part of Model T, tell us only what probabilities to generate. Thus consider that we are trying to generate synthetic traces for a cluster C. The equations can be used to find out which values go into the transition probability matrix M for C. Unfortunately they do not tell us where in the matrix these values go, i.e., they do not tell us $M(u, v)$ for some specific $u, v \in C$. In other words, simply populating the rows of the transition probability matrix in a skewed manner, as described above, is not sufficient. This is because although each row will contain some columns with high values (i.e., from each AP there are certain popular destination APs), these columns could be different for each row (i.e., the popular destinations from each AP are different).

To overcome this problem, we observe that these empirical results suggest we can define a measure of "popularity" of an AP in a cluster by measuring the relative magnitude of the registration traffic for that AP compared to that of the other APs in a given cluster. For a cluster C with intra-cluster transition probability matrix M, we define the set of popular APs as $Pop(C) = \{u \in C : \sum_i N(i, u) > \tau\}$ where τ is a *popularity threshold*. In general we must agree that this part of the model construction is heuristic in nature (for both intra and inter cluster modeling), and given the lack of multiple data sets (e.g., a large number of datasets from multiple campuses) we could not find a statistical technique to approximate this threshold. Nonetheless, we found that using this heuristic provided a reasonable way of capturing and modeling the features of the real data.

The data curve in Fig. 7 uses the average number of incoming transitions to APs as the popularity threshold, i.e., $\tau = \frac{\sum_{i,j} N(i,j)}{|C|}$. On the y-axis it plots the fraction of popular APs, $\frac{|Pop(C)|}{|C|}$, averaged over all clusters of a given size $|C|$, versus the cluster size $|C|$.

The shape of the curve is a direct function of τ. Increasing τ will select only a few popular APs in a cluster and decreasing τ will make most of the APs in a cluster popular. The curve in Fig. 6 will then be shifted down (increase in τ), or shifted up (decrease in τ).

These considerations indicate that the popularity of the APs in a cluster is a salient feature of the data, and that this feature can be used as one of the model parameters. In particular, during trace synthesis, the number of popular APs in a cluster can be used to determine how the transition

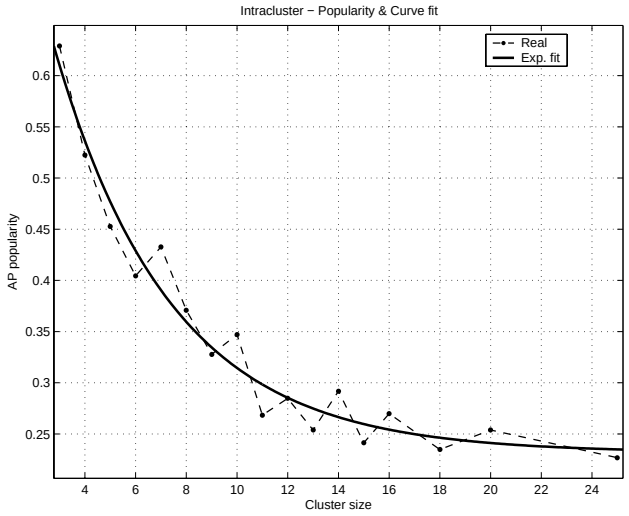

Figure 7: Fraction of popular APs for intra-cluster transitions vs. cluster size.

probability values are assigned to individual cells $M(u, v)$. For example, if the number of popular APs in a cluster of size 10 is 4, then four of the 10 columns in M for that cluster should carry the highest values of transition probability drawn from the Weibull fits.

We capture this notion in the model by fitting a function to the popularity measure in Fig. 7. Different curve fits were tested including polynomial fits, and an exponential (Eq. (3)) described the dependency best according to measures of goodness-of-fit. The parameters of the exponential fit are $p_1 = 0.7261$, $p_2 = 0.217$, and $p_3 = 0.2316$. The goodness of fit measures are $rmse = 0.0233$ and $R^2 = 0.9631$. Using this exponential fit, the numbers of popular APs as a function of cluster size are as follows (listed in {cluster size,number of popular APs} format): {3,2}, {4,3}, {5,3}, {6,3}, {7,3}, {8,3}, {9,4}, {10,4}, {11,4}, {12,4}, {13,4}, {14,4}, {15,4}, {16,5}, {18,5}, {20,5}, {25,6}.

4.3.4 Popular AP assignment

The model for the number of popular APs derived above tells us in the synthesis phase how many APs are to be popular in a cluster. Unfortunately, it still does not tell us precisely which ones. Obviously in a real deployment the popularity of cells is likely to be determined by the real-world activities at those locations (perhaps cafes and bars may be more popular than classrooms or dorm rooms). Clearly some sophisticated data analysis could be applied to try to determine the characteristics of popular APs and their locations. However there is a risk in doing so that the model may become too specific to the data set. Thus, we do not attempt to model this, and simply choose the popular APs at random.

4.3.5 Intra-cluster trace length

Another feature of interest is the distribution of the intra-cluster trace lengths, each represented by the number of transitions executed among the APs of a cluster, between transitions into and out of the cluster. The CDF of the intra trace lengths from the data is shown in Fig. 8. In this plot, as for others in this paper, inset plots are a zoomed-in version of

the larger plot. A fit to this data is determined in the form of a Weibull CDF with parameters $a = 5.227$, $b = 0.295$. The $rmse = 0.01387$ and $R^2 = 0.9269$. This feature will be used for generating synthetic registration traffic.

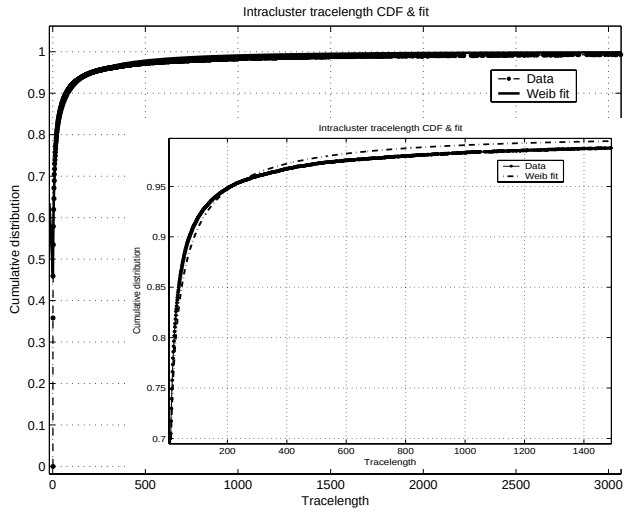

Figure 8: CDF fit for intra cluster trace lengths (inset is a zoomed-in version).

4.4 Inter-cluster Modeling

For modeling inter-cluster patterns we take advantage of the hierarchical nature of the registration patterns, and use the same approach as we took for the intra-cluster modeling. Essentially, we regard each cluster as a virtual AP, and inter-cluster transitions as transitions between these virtual APs. Then we follow the same steps as for a single cluster in the intra-cluster modeling: (1) we derive equations for the transition probability distributions; (2) determine the number of popular clusters; (3) decide how to select the popular clusters; and (4) derive equations for the inter-cluster trace lengths. Since there is only one cluster of virtual APs, reducing the number of model parameters, as we did for the intra-cluster modeling case, is not required.

The empirical transition probability CDF and the corresponding distribution fit for these probabilities is shown in Fig. 9. The fit is a Weibull distribution with parameters $a = 0.0006967$, $b = 0.2673$. The $rmse = 0.01755$ and $R^2 = 0.9873$.

The number of popular clusters is found using the popularity threshold method used above for intra-cluster transitions. However, we found that with the threshold set to be the average cluster popularity there were a large number of clusters selected. In other words, cluster popularity is even more skewed than AP popularity within clusters. We thus set the threshold to be $\tau = 50\%$ of the maximum number of inter-cluster transitions. With this the number of popular clusters was found to be 2. Once again, we choose the actual popular clusters randomly.

Finally, the CDF corresponding to the empirical inter-cluster trace lengths obtained from the data is shown in Fig. 10. A fit to this distribution is determined in the form of a Weibull CDF with parameters $a = 351.8$, $b = 0.8815$. The $rmse = 0.01346$ and $R^2 = 0.9973$.

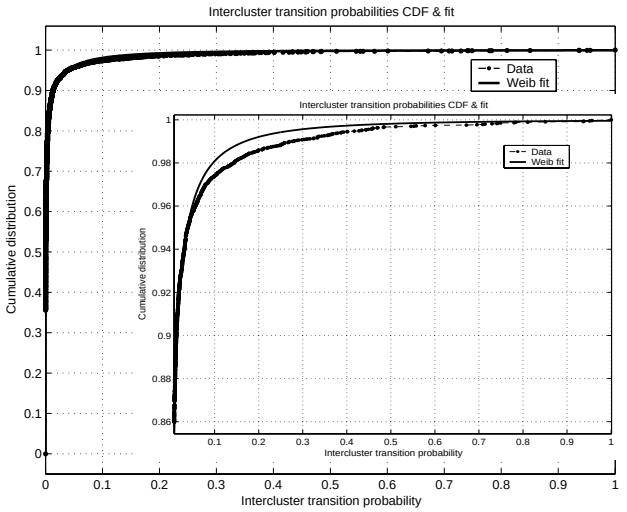

Figure 9: Empirical CDF and fit for inter-cluster transition probabilities.

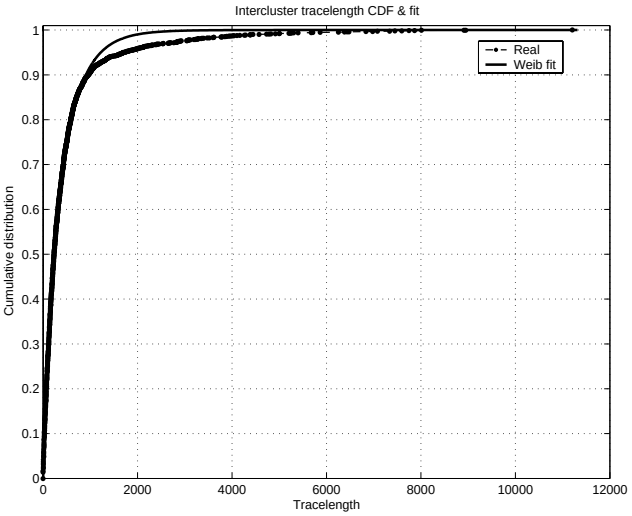

Figure 10: Empirical CDF and fit for inter cluster trace lengths.

To summarize, Model T abstracts the entire set of traces (over 6000 users for 2 years) into a relatively small set of parameters which are collected in Table 3.

5. MODEL EVALUATION

In this section we present a comparison of Model T, which has been developed using the training set of experimental data, with the test set of experimental data. We also compare it with a modified random waypoint model.

The basic input to the model is the total number of mobile users n, and the total number of APs m. Optionally, additional information can be supplied, such as the number of clusters and their sizes, the initial placement of mobile users among the m APs, etc. In the following we describe the general case when that information is not available; the cases where it is should be clear.

If the partitioning of m APs into a number of clusters c and their sizes $\{C_i: i = 1 \ldots c\}$ is supplied, we initialize the clusters accordingly. Otherwise, the model uses the cluster distribution observed in the Dartmouth data, illustrated in Fig. 2. Obviously, if the model user provides the cluster distribution greater independence from the experimental data set is obtained, at the cost of additional effort.

Table 3: Summary of model parameters

Params.	Symbol	Distribution	Values	GOF
No. of users	n	User supplied		
No. of APs	m	User supplied		
No. and distrib. of clusters	c_i, number of APs in cluster i	Case (a): User supplied		
		Case (b): From CDF Weibull Eq. (1)	a =6.561 b =1.253	rmse= 0.03892 R^2 =0.9858
Number of popular APs in a cluster of size c	Pop(c) of APs in cluster c	Exponential: Eq. (3)	p_1 =0.7261 p_2 =0.217 p_3 =0.2316	rmse= 0.0233 R^2 =0.9631
Intra-cluster transition probability	$M_c(i,j)$ where M_c is the intra prob matrix, c is cluster size, i,j are the APs in c	Weibull: params. a, b distributed according to an exponential Eq. (3)	p_{a1} =2.55 p_{a2} =0.5437 p_{a3} =0.0264	rmse= 0.0259 R^2 =0.9688
			p_{b1} =1.211 p_{b2} =0.3802 p_{b3} =0.3552	rmse= 0.115 R^2 =0.5112
Intra-cluster trace length	l_{intra}	Weibull	a =5.227 b =0.295	rmse= 0.01387 R^2 =0.9269
Inter-cluster transition probability	$M(i,j)$ where M is the inter prob matrix, i,j are the clusters (virtual APs)	Weibull: params. a, b	a =0.0069 b =0.2673	rmse= 0.01755 R^2 =0.9873
Inter-cluster trace length	l_{inter}	Weibull	a =351.8 b =0.8815	rmse= 0.01346 R^2 =0.9973

5.1 Trace generation

5.1.1 Empirical model traces

We generate synthetic traces based on the empirical model as follows. Each trace consists simply of a sequence of APs a_i representing the AP visited at step i.

If the initial placement of the n users at the m APs is provided, we utilize that. Otherwise, Model T simply places users at APs randomly. We have observed in the Dartmouth data that the initial placement is in fact highly skewed; thus, a more sophisticated model could use the placement distribution of the Dartmouth data to generate an initial placement; we discuss this point later.

Let us first describe the generation of an intra-cluster trace in a given cluster of size c (as defined earlier). For each trace, a transition probability matrix M is created, row by row, by drawing values from the model transition probability CDF for a cluster of size c.

Suppose the current AP in the trace generated so far is AP i and we want to find the next AP that should appear in the trace. The row $M[i]$ contains the transition probabilities to other APs as discrete values. Let $M'[i]$ be $M[i]$ sorted in decreasing order of transition probability, and let $M''[i]$ be the cumulative value of $M'[i]$, i.e., $M''[i] = \sum_{j=0}^{i} M'[j]$. We generate a random number $y \in [0,1]$, and find the closest index l such that $M''[l]$ is the nearest value to y, with ties broken arbitrarily. Let $x = M''[l] - M''[l-1]$. The next AP is then AP j where $M[j] = x$. As far as the length of the trace is concerned we determine that from the intra-trace model CDF previously discussed.

To generate an inter-cluster trace, we consider the clusters as virtual APs and repeat the process described above, but with the corresponding inter-cluster model parameter values described in the previous section. Starting from a random cluster, a transition is made to other clusters based on the inter-cluster transition probability matrix. Once a cluster is selected a number of intra-transitions are made (the number which is obtained from the intra-cluster trace length CDF fit, Fig. 8) before moving to another cluster. This process is repeated for all users to generate the synthetic traces.

5.1.2 Random waypoint traces

For comparison purposes we also generate traces for a theoretical model, namely random waypoint. We actually consider a modified version of the traditional random waypoint model, in that we do not jointly model the speed and the AP residence time features. The reason is that the random waypoint model is for geographical mobility, whereas we need to model registration patterns. There is no widely-accepted model of user registrations, so we chose to use this modified version of a well-known model for geographical mobility instead. We call the modified random waypoint model RW. Obviously more complex theoretical models could also be used, but the salient differences between our empirical model and simple theoretical models can be illustrated using random waypoint alone.

We compare our intra and inter cluster model to the RW model by generating RW intra- and inter- cluster traces with lengths equal to those of the real data. For a given cluster, to generate an intra-cluster RW trace, we take the number of transitions to be the same as that of the real data, and make that many random moves among APs in the cluster. This process is repeated hierarchically to generate inter-cluster

RW traces, but now considering each cluster to be a virtual AP.

On the other hand, to compare entire traces, RW traces are generated without clustering. Rather, the AP transitions are selected at random among the set of all available APs. The trace length is taken to be the same as that of the real data. It is possible to generate a hierarchical RW model, where the APs are clustered, an inter-cluster RW is applied, and for each cluster selected, an intra-cluster RW is used. Nevertheless, even if this method were applied, it would still not capture the skewness in transition probability that we observe at both inter- and intra-cluster level, and thus in general we would not expect a major improvement in the fidelity.

5.2 Metrics

We need metrics to quantify how well Model T, which is based on the training data, captures the features of the test data, and how it differs from the RW model.

In the following we describe results for a series of metrics that examine different facets of the model. Each metric is calculated for all traces in the test data (called the "Real" or "Data" traces), the traces generated by Model T (called the "Synthetic" traces), and the traces generated by the RW model (called the "random waypoint (RW)" traces).

Metrics that are calculated for intra-cluster transitions operate on each trace (whether it is a Real, Synthetic or RW trace) by considering only the transitions in that trace that occur between APs of the same cluster, ignoring OFF states. Inter-cluster metrics are calculated by considering only the transitions in a trace that occur between APs of different clusters, ignoring OFF states. Finally where relevant we also calculate metrics for the total trace, which includes both intra- and inter-cluster transitions.

A synthetic inter-cluster trace consists of a sequence of cluster IDs. Thus, the number of synthetic traces corresponding to a particular cluster (intra-cluster traces) is equal to the number of times that cluster ID occurs in the inter-cluster trace. The total number of synthetic traces for a *cluster size* is obtained by summing the number of traces generated for each cluster of that size, for all users. Multiple generations of synthetic traces were done, and the variation of the metrics was very small, such that we have not quantified it. For RW, the number of traces generated for a cluster is the same as for the real data.

We first describe the results for simple metrics that deal with AP popularity or user trace characteristics. We then describe results for somewhat more complicated metrics that attempt to compare Model T with the test data and with RW on a more global level.

5.3 Metric M1: Popular APs

We start with a simple metric that reflects one of the key features captured in Model T, namely AP popularity. Recall that AP popularity information was used as an input to the model; however, the input was based on the training data. Using it as a metric on the output simply allows us to do a sanity-check that the model does capture this feature for the test data. It also allows us to compare the model to RW.

The AP popularity metric M1 calculates, for each cluster C, the fraction of popular APs, $\frac{|Pop(C)|}{|C|}$, and averages it over all clusters of size $|C|$; an AP is in $Pop(C)$ if the total number of incoming intra-cluster transitions to the AP in the trace exceeds the average number for all APs in that cluster. That is, if we denote the *average popularity* of an AP in a cluster C as

$$AvPop(C) = avg_{u \in C}(\sum_i N(i,u)) \quad (4)$$

then

$$M1(c) = avg_{|C|=c} \frac{|\{u \in C : \sum_i N(i,u) > AvPop(C)\}|}{|C|} \quad (5)$$

The values of M1 for Real, Synthetic and RW are shown for different cluster sizes in Fig. 11. Clearly the agreement between the Real traces using the test data and the Synthetic traces based on the training data is very good. For both Real and Synthetic traces M1 decreases as cluster size increases, reflecting the fact that the number of popular APs does not change much as the cluster size increases. Also, RW diverges from the other two traces, hovering around the value 0.5, as expected. For small clusters RW is closer to the other traces because the number of popular APs becomes around half the cluster size. For large clusters we observe that the Synthetic traces have slightly fewer popular APs than the Real traces, although the agreement is still very good.

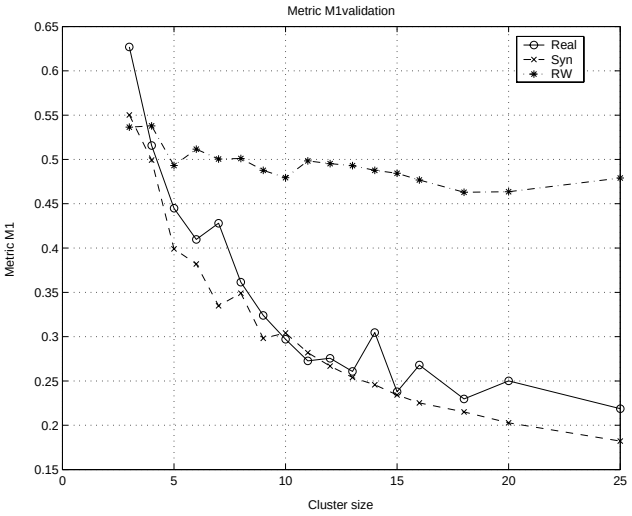

Figure 11: Popular APs by metric M1 for intra-cluster transitions.

5.4 Metric M2: Peak-to-average popularity

While M1 indicates the fraction of popular APs, M2 attempts to capture the degree of skew in AP popularity. M2 calculates the ratio of the number of incoming intra-cluster transitions to the most popular AP in the cluster to the average number of incoming transitions for all APs in that cluster. More precisely,

$$M2(c) = avg_{|C|=c} \frac{max_{u \in C}(\sum_i N(i,u))}{avg_{u \in C}(\sum_i N(i,u))} \quad (6)$$

The values of M2 for Real, Synthetic and RW are shown for different cluster sizes in Fig. 12. Once again the agreement between the Real traces using the test data and the Synthetic traces based on the training data is very good. For both Real and Synthetic traces M2 increases as cluster size

increases, reflecting the fact that for larger clusters many of the APs have low popularity, thus reducing the average AP popularity. This is consistent with the observations for M1. Clearly, RW does not capture the popularity skew in the data at all.

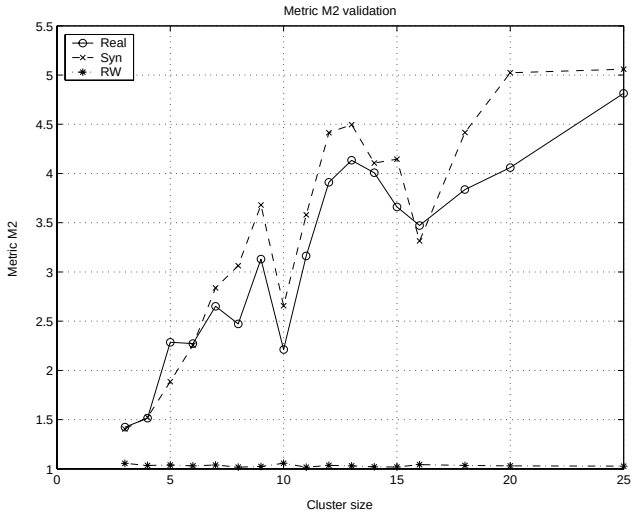

Figure 12: Peak-to-average popularity by metric M2 for intra-cluster transitions.

We also calculate M2 for inter-cluster transitions only, and find that while the Real and Synthetic traces are close, the RW traces are significantly different (see Table 4). Finally, we calculate M2 over the entire trace, and show similar results (Table 4).

Table 4: Metric M2

	Metric M2	
	Inter	Total
Real	14.094	32.7593
Synthetic	17.499	45.9284
RW	1.017	1.0404

5.5 Metric U1: Number of distinct APs

Metrics M1 and M2 describe the traces from a per-AP point of view. In metrics U1 and U2 we consider an orthogonal facet of the data, the per-user point of view.

Metric U1 is the number of distinct APs visited by a user over its entire trace. This helps to see if the Model captures locality of user transitions. Note that this differs from the notion of popularity expressed in M1 in that one could have a very popular AP, which is thus visited frequently by many users, but the users could then visit a large number of other APs in a uniformly distributed fashion. Metric U1 shown in Fig. 13 shows that this is not the case in the data, and Model T captures that reasonably well.

5.6 Metric U2: Percentage of inter-cluster transitions

Metric U2 is the percentage of inter-cluster moves in the total trace for a user. Recall that aggregated over all users this value is only about 25%. Metric U2 gives a finer-grained breakdown of this characteristic, and shows that the distribution of intra versus inter cluster transitions in the data,

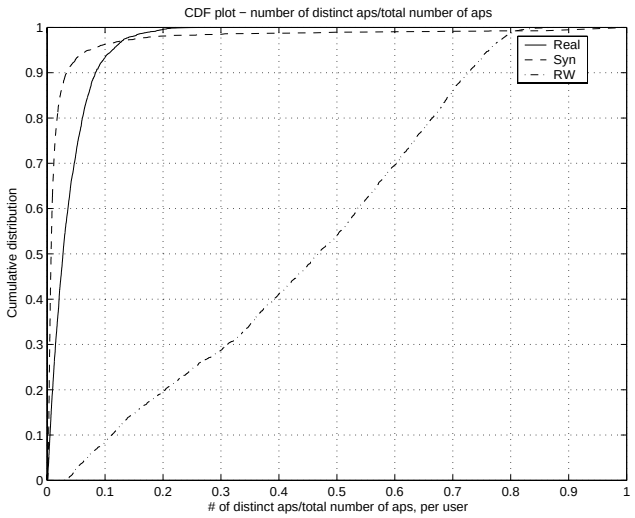

Figure 13: CDF of metric U1: number of distinct APs visited by a user.

on a per-user basis, is captured reasonably well at this fine grain by the model. The metric U2 is illustrated in Fig. 14.

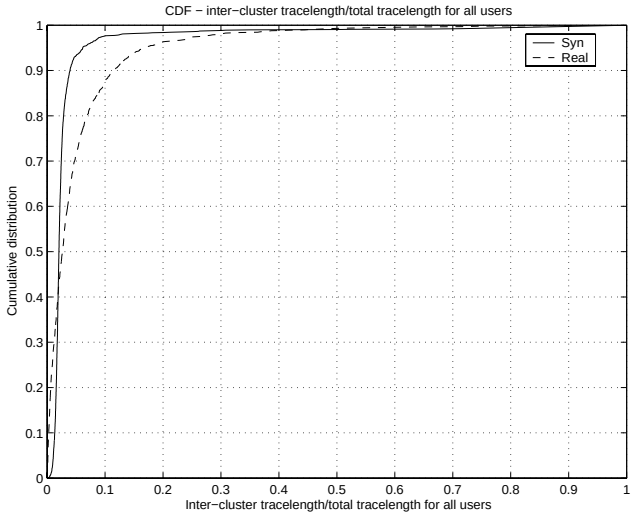

Figure 14: CDF of metric U2: fraction of inter-cluster transitions for a user.

5.7 Metrics L1, L2: Transition probability matrix norms

So far we have considered the behavior of the traces either from a per-AP or a per-user perspective. We now introduce a metric that considers the overall movement pattern of users between APs, by looking at the transition probability matrix as a whole.

We generate the transition probability matrix Re for the training data, and the corresponding matrices Syn and RW derived from the Synthetic traces and the RW traces. Note that we do not simply generate Syn and RW directly; we actually generate traces, in accordance with Model T or the

random waypoint model, and derive the observed transition probability matrices. This allows us to capture, for example, the fact that in Model T trace length is derived from a curve fit to the training data, the start AP in a trace for a given user is randomized, etc. – in other words, capture all that goes into the model.

We then calculate how much *Syn* and *RW* differ from *Real* by calculating two matrix difference quantities, which we call the *L1 norm* and *L2 norm*. These are defined as follows for arbitrary matrices A and B where subscript C denote that the matrix is for a given cluster C. For inter-cluster trace and total-trace norms there is only one "cluster".

$$L1(c) = avg_{|C|=c} \frac{1}{c^2} \frac{\sum_{i,j} |A_C(i,j) - B_C(i,j)|}{\sum_{i,j} |A_C(i,j)|} \quad (7)$$

$$L2(c) = avg_{|C|=c} \frac{1}{c^2} \sqrt{\frac{\sum_{i,j} |A_C(i,j) - B_C(i,j)|^2}{\sum_{i,j} |A_C(i,j)|^2}} \quad (8)$$

While the two norms are similar, in some cases they can show different results as, in general, the L2 norm tends to mask the effect of outliers. For our purposes we check both norms. In Figs. 15, 16 we plot the L1 and L2 norms for the difference between Model T and the test data (denoted "Re-Syn"), as well as the difference between RW and the test data, as a function of cluster size. We see that both the L1 and L2 norms are very small when comparing the test data with Model T, and that in general RW deviates very significantly from both.

We observe that, since we have only one cluster corresponding to size 10 (compared to at least 2 and typically 5 for all other sizes), the metric is more noisy for this particular case.

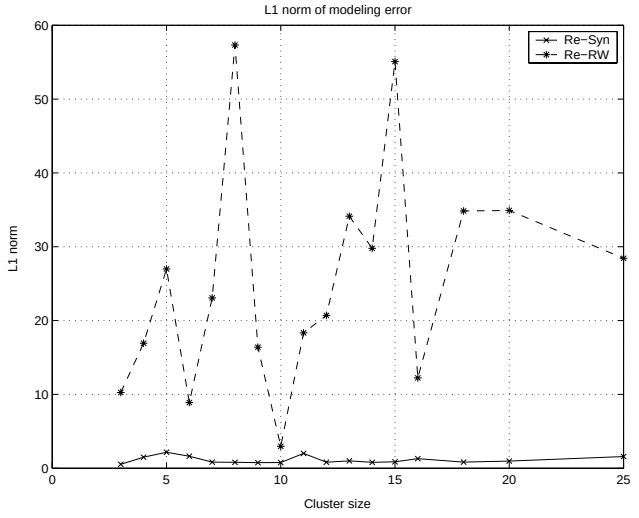

Figure 15: L1-norm of transition probabilities modeling error for intra-cluster case.

In Table 5 we show the corresponding values when considering only inter-cluster transitions, and for the complete traces. Once again we see that Model T is close to the test data by this global registration pattern metric.

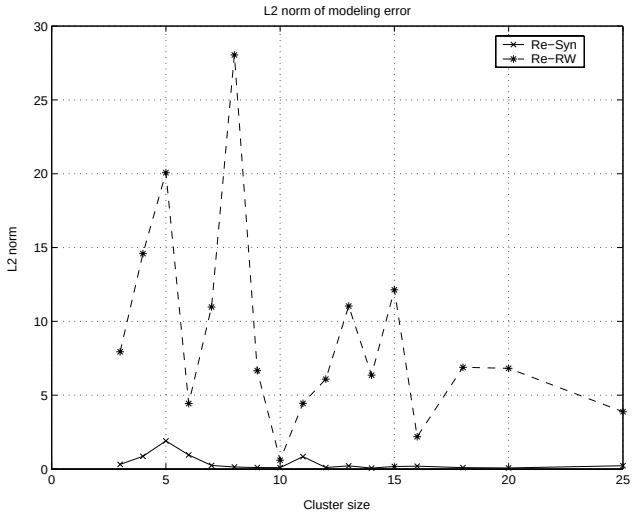

Figure 16: L2-norm of transition probabilities modeling error for intra-cluster case.

Table 5: Norms of modeling error

	L1-norm		L2-norm	
	Inter	Total	Inter	Total
Real-Synthetic	0.6549	0.1707	0.0123	0.00426
Real-RW	23.516	3.0547	1.3912	0.2451

6. MODIFYING MODEL PARAMETERS

In this section we illustrate how a user of the model can modify the parameters of the model, i.e., "turn the knobs" on Model T.

Consider a scenario where the user of the model, Alice, has generated registration patterns with Model T, fed them into her simulation model (a simulation that tries to assess the signaling overhead of the Mobile IP protocol), and obtained the results she is interested in. Now she would like to see what would be the result if the registration patterns were not as skewed as given by Model T, i.e., if they were closer to a random waypoint model. On the other hand, Alice does not want patterns that are completely random, as they may not be very realistic. (Such a study might be very useful, for example, if Alice suspects that her own mobility management protocol Alice-MIP, will perform better than standard Mobile IP in that case.)

Model T accommodates Alice by allowing her to vary the parameters a and b of the Weibull fits of the transition probability distributions. In particular, she simply decides to set them to fixed values $a = 0.4$ for all clusters, and $b = 0.98$ for all clusters; clearly they could be varied in a more sophisticated manner for different cluster sizes.

We generate traces for Model T$'$, which is Model T with these new transition probability parameters, and gauge the effect using some of the metrics defined in the previous section. Fig. 17 shows the effect on M1, which reflects the number of popular APs. Comparing it with Fig. 11 we see that the traces for Model T$'$ fall between the training data and RW. Similar results are shown in Figs. 18, 19, 20 for intra-cluster metrics M2 and norms L1 and L2; these can

be compared with Figs. 12, 15, and 16 respectively. Inter-cluster and total-trace metrics show similar behavior and are omitted for brevity.

Thus Alice can modify the degree of closeness to the training data by changing a and b, i.e., "turning the knob" on the model. Obviously, there are limits in that the Weibull distribution cannot be made to exactly replicate a random distribution. Nonetheless, Model T gives a degree of flexibility that would be hard to obtain if the experimental data was used as is.

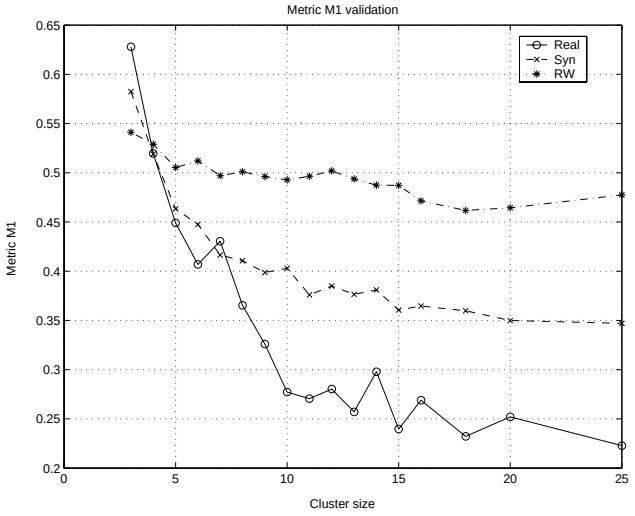

Figure 17: Popular APs by metric M1 for the modified model.

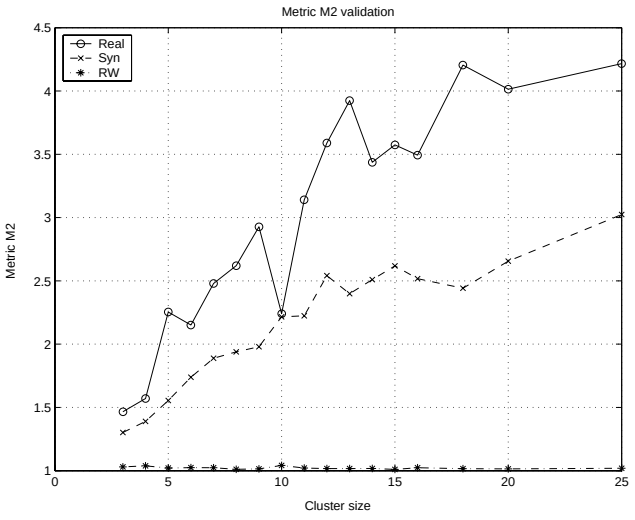

Figure 18: Peak-to-average popularity by metric M2 for the modified model.

7. DISCUSSION

In this section we discuss several aspects of the model, in particular variations that did not work, as well as the incorporation of time.

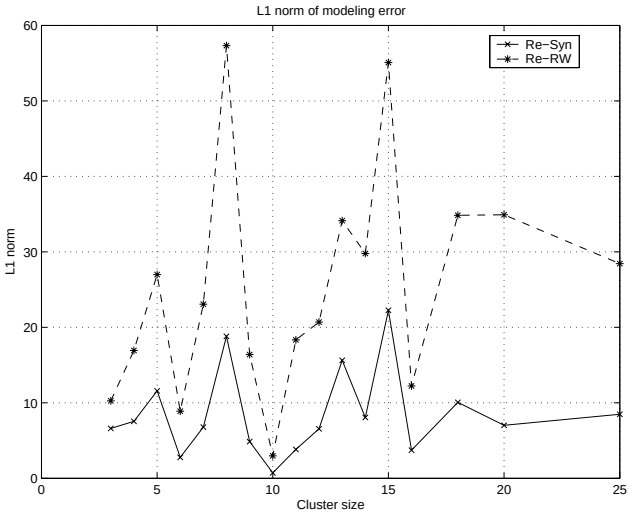

Figure 19: L1 norm for the modified model.

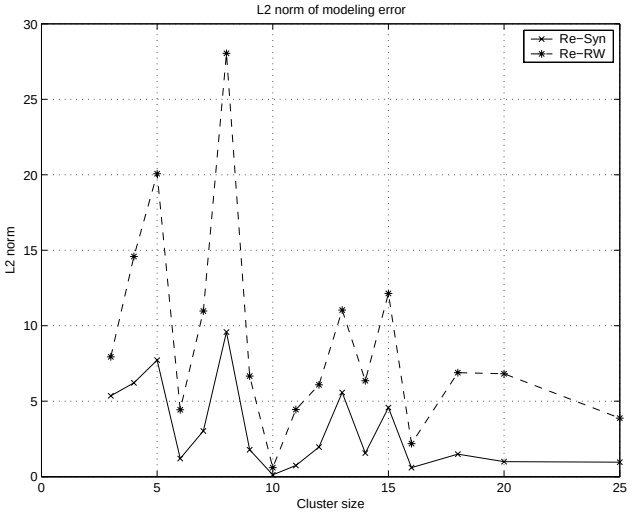

Figure 20: L2 norm for the modified model.

7.1 Model variations

Discussion of the following model variations may be useful to other researchers and also help illustrate some of the non-trivial subtleties in the model.

7.1.1 Dependence on distance

When developing the model of the transition process from one AP to another, we tested the hypothesis that inter-AP distance plays a major role in determining this process. In other words, the likelihood of transition from an AP to another increases with decreasing inter-AP distance. If true, this feature could have been used in the model design.

We note that in our experimental data, as mentioned in section 3, we did not have the geographical location for about 100 of over 500 APs. So this hypothesis could only be tested for the APs for which geographical coordinates were available. However, for the remaining data, which is still quite substantial, we found this hypothesis to be unreliable; while there was some dependency in the aggregate, it was

weak. Specifically, we investigated the dependency between the inter-AP transition probabilities for the training data, i.e., $M(i,j)$ between APs i and j (as defined in section 5), and the inter-AP distance $d(i,j)$.

We calculated the inter-AP distance $d(i,j)$ as a Manhattan distance as well as an Euclidean distance, and found that neither distance metric showed a correlation with inter-AP probability, although the Manhattan distance was slightly better. An example of the relation between $M(i,j)$ and $d(i,j)$ is illustrated in Fig. 21.

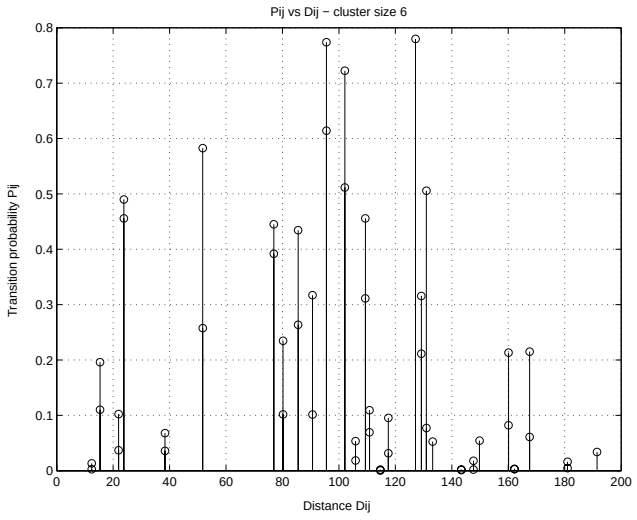

Figure 21: Sample interAP transition probability vs. inter-AP Manhattan distance in cluster of size 6.

7.1.2 Trace generation

Another issue worth noting is in how synthetic traces are generated once the transition probabilities have been found. Consider the case of intra-cluster traces only, for simplicity. Suppose that for a cluster C the equations for the transition probabilities based on fitting the training data CDF have been found. One intuitively attractive possibility is as follows, which we call *static matrix*. First populate a transition probability matrix with entries $M(i,j)$, where each entry is drawn appropriately from the transition probability equations. Then, for each user, generate transitions from one AP to the next in accordance with the values in M.

An alternative is to pick fresh transition probabilities from the equations *on the fly* as each user trace is generated. If the user is at AP i, populate the row $M(i)$ from the transition probability equations, and choose the next AP based on the highest value in that row. It turns out in our experiments that the on-the-fly method works much better in generating synthetic traces that come close to the training data. This is because it picks far more values from the transition probability equations than the static matrix method, and hence achieves a better overall statistical sample.

7.1.3 Selecting the start AP

Currently Model T makes the assumption that each user trace starts at a random AP, as noted earlier. Thus a straightforward improvement to Model T would be to capture the distribution of start APs.

To test the value of this, we generated a complete set of synthetic traces where the start AP for each trace was simply set to be the same as for the test data, i.e., an extreme case where the model perfectly captures the start AP distribution. We found that this made no discernible difference in the quality of the synthetic traces generated, at least in terms of the metrics we defined. This is obviously because, even though trace lengths are unevenly distributed they are long enough that the start AP makes no difference for our metrics.

7.2 Incorporating time

This paper focuses on the task of building a spatial model of user registration patterns, and does not explicitly include the time at which a transition occurs or the residence time at an AP.

It is straightforward to incorporate an *independent* time model into Model T. For example, suppose one wanted to use a random-walk style of time model. The residence time of the user at each AP is then drawn from a uniform distribution over a time interval $[t_{min}, t_{max}]$, with the endpoints of the interval chosen based on domain information, or as free parameters whose impact can be examined; the same (or a different) interval is used for the OFF state. It is clear that this time model can be made almost arbitrarily sophisticated. In fact this approach is similar to that taken by geographic mobility models, such as random walk and random waypoint [1], although they may choose geographical velocities of motion rather than time intervals. Model T can clearly use this approach in a trivial way.

Another approach that is driven by the empirical data is to fit mean residence time, mean OFF time and similar distributions to the Dartmouth data, and use these distributions in the independent time model. Thus we can take the information supplied by analysis such as [4] and again use it in a straightforward way in conjunction with Model T.

However, we observe that while using an independent time model will give traces that match the time values in the data in an *independent* sense, this approach will not capture the interactions between time and AP locations. For example, suppose the data shows that popular APs can be roughly divided into two classes: those with very short residence times (e.g., entryways to buildings) and those with very long residence times (e.g., cafes). Using an independent time model will pick residence times from the same distributions for both classes of APs.

Modeling the interaction between the spatial and temporal patterns in user registrations is a challenging problem that we leave for future work. For the moment we note that Model T can easily be augmented with an independent time model as described above; we omit this exercise in this paper.

8. CONCLUSIONS AND FURTHER WORK

This paper reports work on developing Model T, an empirical user registration model capturing key spatial features of registration patterns; it does not attempt to model geographic mobility. While the model can easily be augmented with an independent time model, this paper does not focus on illustrating that process.

The key features of the empirical data that Model T captures and utilizes for synthetic trace generation are:

- User registration patterns show a distinct hierarchy. APs can be clustered based on the probability of inter-AP transitions
- Cluster sizes are highly skewed, with many small clusters (5 APs per cluster or less), some medium clusters (6 - 15 APs), and a few large clusters (more than 15 APs)
- Within each cluster the transition probabilities are highly skewed and can be fit to a Weibull distribution
- Intra-cluster transition probabilities are relatively well-behaved in that the variation of the parameters of the Weibull distributions for different cluster sizes can be fit to exponential distributions
- The fraction of popular APs within a cluster for different cluster sizes, as defined by a popularity threshold, can be fit to an exponential distribution
- Assigning which APs are popular within a cluster can be done in a random manner, with no significant effect on our metrics
- The number of transitions within a cluster (i.e., intra-cluster trace length) are also highly skewed and can be fit to a Weibull distribution
- There is similarity across hierarchies, i.e., inter-cluster registration patterns generally have the same characteristics as intra-cluster patterns, and similar distributions can be used

The model presents several insights that have not been captured in existing models or in existing analyses of campus WLAN data. For example, we find registration patterns show a distinct hierarchy, with highly skewed cluster popularities, and model this feature. This is not captured in the random waypoint model. Such an insight can be used, for example, as follows. When evaluating a new protocol design, a model that captures the fact that registrations are skewed and exhibit locality may favor certain types of protocol designs (e.g., those that rely on caching of user registrations) over others. In fact this can also be a guide or influencing factor for protocol design itself. As another example, the existence of clusters may motivate, in some situations, protocols that are oriented towards electing local cluster heads which perform local control and signaling functions.

Almost all the curve fits that Model T uses for the empirical data have very good fits, with very high R^2 values and low $rmse$ values. We evaluate Model T with six metrics, two which investigate how well it captures the per-AP behavior it tries to represent, two that investigate how well it captures per-user behavior, and two that look at the global distribution of transition probabilities. For these metrics Model T, which is derived based on a training data set, is used to generate synthetic traces that are found to agree very well with the test data set. In addition, it is clear that a random waypoint model is not at all representative of the empirical data.

We show how the user of Model T can modify its parameters and illustrate the ease and effect of doing so for an example scenario. We also briefly discuss variations of Model T.

Clustering plays an important role in Model T. However, we did not a priori assume clusters in the data, but in fact attempted to mine the data and discover if clusters existed or could be inferred. The MCL tool takes a particular approach to determine clusters which seems very suitable for our problem domain as it is naturally modeled in terms of graphs. We found that using the clusters generated by MCL allowed us to get a very succinct and reasonable characterization of the data, much more than without using the clusters. In fact, using the clustering helped us to structure the model well while still retaining fidelity to the empirical data in terms of the metrics we studied. Thus in general we believe that clustering exists and is a *real* feature of the data. Nonetheless, we recognize that the specifics of the clustering are dependent on the actual clustering algorithm and tool used. Trying a variety of other clustering algorithms and techniques is an interesting subject for future study.

Another challenge for further work is the development of a joint empirical spatial-temporal registration model. We believe the methodology we have developed could also be applied to similar data sets from other campuses; doing so is outside the scope of the present paper but an interesting item for further work. It would also be interesting to generate models similar to Model T and apply them for scenarios where there are "bazaars" of multiple coexisting access networks and third-party services, as we expect in the future [11]. Finally, a comparison of Model T with a suite of other models such as the obstacle model would be a substantial additional study and it is left for further work.

Acknowledgements. We are indebted to David Kotz, Tristan Henderson, and their colleagues at Dartmouth College for making their experimental data available to us as well as providing additional information (e.g., AP locations) and assistance.

9. REFERENCES

[1] T. Camp, J. Boleng, and V. Davies, "Mobility models for ad hoc network simulations," *Wireless Communication and Mobile Computing*, vol. 2(5), pp. 483–502, 2002.

[2] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri, "Towards realistic mobility models for ad hoc networks," in *Proceedings of MobiCom*, Sep. 2003.

[3] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," in *Proceedings of MobiCom*, Sep. 2002.

[4] R. Jain, A. Shivaprasad, D. Lelescu, and X. He, "Towards a model of user mobility and registration patterns," *Mobile Computing and Communications Review, ACM SIGMOBILE*, vol. 8(4), pp. 59–62, 2004.

[5] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of Infocom*, Mar. 2003.

[6] F. Chinchilla, M. Lindsey, and M. Papadopouli, "Analysis of wireless information locality and association patterns in a campus," in *Proceedings of Infocom*, 2004.

[7] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *Proceedings of MobiSys*, May 2003.

[8] D. Tang and M. Baker, "Analysis of a local-area wireless network," in *Proceedings of MobiCom*, Aug. 2000.

[9] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive Wi-Fi mobility data," in *Proceedings of Infocom*, Mar. 2004.

[10] S. van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, University of Utrecht, May 2000.

[11] R. Jain, "Towards 4G: From Cathedral to Bazaar," Panel talk, IEEE Broadnets Symp., Oct. 2004. See http://www.docomolabs-usa.com