Building Realistic Mobility Models from Coarse-Grained Traces

Jungkeun Yoon, Brian D. Noble, Mingyan Liu
Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122
{jkyoon, bnoble, mingyan}@eecs.umich.edu

Minkyong Kim
Department of Computer Science
Dartmouth College
Hanover, NH 03755
minkyong@cs.dartmouth.edu

ABSTRACT

In this paper we present a trace-driven framework capable of building realistic mobility models for the simulation studies of mobile systems. With the goal of realism, this framework combines coarse-grained wireless traces, i.e., association data between WiFi users and access points, with an actual map of the space over which the traces were collected. Through a sequence of data processing steps, including filtering the data trace and converting the map to a graph representation, this framework generates a probabilistic mobility model that produces user movement patterns that are representative of real movement. This is done by adopting a set of heuristics that help us infer the paths users take between access points. We describe our experience applying this approach to a college campus, and study a number of properties of the trace data using our framework.

Categories and Subject Descriptors

I.6 [**SIMULATION AND MODELING**]
; G.3 [**PROBABILITY AND STATISTICS**]

General Terms

Measurement, Experimentation

Keywords

Statistical Mobility Model

1. INTRODUCTION

Mobility models are an indispensable tool in simulating mobile systems. They supply the movements for each node, allowing one to examine a wide variety of patterns and behaviors. However, the vast majority of models have little or no relevance to *real-world* movements. The most common example, Random Waypoint [15, 8], models nodes with completely random destinations in a bounded, open area—not at all similar to the way real people move.

While there have been some important strides in improving this state of affairs, we are not aware of any modeling approach that is able to faithfully represent real movement. We believe this is due primarily to the difficulty of capturing such movement. For example, one could recruit a set of users to wear or carry a location-tracking device, such as a GPS receiver or a PlaceLab [4] client, but building a large enough user community for a meaningful model would be challenging at best. Furthermore, many users are uncomfortable having their specific locations tracked and reported for any significant period of time. Alternatively, one could passively monitor a physical space, watching and recording movement, but this is both time consuming and cumbersome.

In this paper, we present a system that is able to derive a representative, realistic mobility model by combining *coarse-grained* wireless traces—association data between WiFi clients and access points—with a map of the measured space. The map is first converted to a graph representation, with known locations for each measured access point. The wireless traces are then filtered, giving us at minimum a sequence of starting and ending points for *trips* taken by each user. Users with devices that are always on can yield even more movement data.

The map combined with a set of heuristics allows us to model the user movement as a second-order Markov chain and generate from the filtered data trace a set of *transition probabilities* from one map location to another. Specifically, the heuristics generate candidate routes between locations, biasing for distance, and constrained by map features; transition probabilities are then obtained by aggregating across such routes. The model is initially populated by estimating user densities at each map location, also obtained from the data trace.

We have applied our scheme to a college campus. We evaluated the quality of the derived model by comparing the transition probabilities computed at a variety of intersections with actual observations of pedestrians at those locations. While our model does not perfectly capture these counts, individual predictions made by the model show high correlation and low error when compared to actual traffic—in short, coarse-grained data gives us surprisingly representative results.

This modeling framework carries with it several advantages:

- a statistical mobility model that is based on *real* movement, not a random distribution,
- it requires only coarse-grained data collection in the form of AP associations—a data set relatively easy to

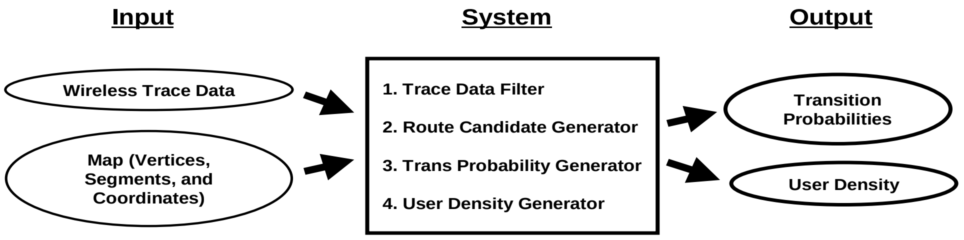

Input
System
Output
Wireless Trace Data
Map (Vertices, Segments, and Coordinates)
1. Trace Data Filter
2. Route Candidate Generator
3. Trans Probability Generator
4. User Density Generator
Transition Probabilities
User Density

Figure 1: Statistical mobility model (SMM) generating system overview.

obtain—but is able to generate representative movement at much finer grain by combining a physical map with reasonable heuristics,

- the coarse-grained collection imposes a relatively modest privacy intrusion, and
- the models can be updated easily by using recent trace data.

If one draws from longitudinal data, the resulting model captures aggregate characteristics, but not diurnal effects. By considering only subsets of the trace data—for example weekday mornings"—one can also tailor models for a specific time window of interest.

Realistic, site-specific mobility models have several interesting applications beyond serving as a basis for mobile system simulation. For example, urban planning projects depend on measuring traffic through various points; doing so through first-hand observation is very expensive. Likewise, socially-based games and augmented reality applications depend on physical movements in a real setting.

The remainder of this paper is organized as follows. Section 2 briefly covers related work on mobility models so far. Section 3 describes how to generate our model, while Section 4 shows the evaluation of our model by comparing the results to the real measurement. Furthermore, we apply our model to a larger size of trace data for in-depth analysis on mobility in Section 5. Section 6 concludes this paper.

2. RELATED WORK

Most mobility models used in simulation studies have been synthetic random mobility models [8, 23, 9, 13, 22, 5, 26, 21, 7, 18]. While they differ in details, each follows a similar pattern. Given some idealized, typically open, space, each of a number of principals independently repeatedly chooses a destination or direction and speed or transit time, at random, from some probability distribution. Some of these [19] augment this basic approach with a stochastic selection of various classes of mobility to enrich the model. These models have several advantages: they are simple, and it is easy to compare different protocols and systems using them. However, they all suffer from the same problem—they may or may not capture the way people move in realistic spaces. Worse, one may be led to incorrect conclusions based on these models, as has been shown in the literature [26, 21, 7].

To overcome the lack of reality, Jardosh et al. [14] implemented a mobility model considering obstacles (e.g., buildings) that block the signal propagation. This provides a better model of space, but may not significantly increase the realism of movement in that space. On the other hand, Herrman [11] and Musolesi et al. [20] proposed two different social mobility models considering the interactions and relationships between mobile users. Furthermore, Tan et al. [25] collected the real trace data from a networked first-person-shooter game and built a modified version of random waypoint model for networked games. These improve plausibility, but still may or may not generate models representative of any particular physical space.

One could presumably record the precise movements of individuals and derive more accurate models than our own. However, we are unaware of any study that has done so. Such a study would either require significant personnel resources to observe a large number of locations, or a fine-grained, large-scale, ubiquitous location-tracking infrastructure; either is prohibitively expensive. Our contribution provides a midpoint between purely hypothetical models and fine-grained observations; we generate plausible models that adequately match real behavior, without significant costs or infrastructure beyond common wireless access deployments.

3. GENERATING THE STATISTICAL MOBILITY MODEL (SMM)

In this section we describe the design methodology used to produce statistical mobility model (SMM). At the high level, our method is a trace-driven one in that it takes as input real movement trace data and produces as output a statistical mobility model, which generates synthetic movement patterns that are statistically similar to the trace data. This model is built offline using trace data, and can be used to generate movement for use in a simulation study.

There are two important features of our approach. First, it takes as input not only the trace data, but also a specific topological map on which the trace data were collected. Consequently the statistical mobility model generated by this method is also within the context of a given map. While the applicability of the resulting mobility pattern is limited to this particular topological environment, our methodology is much more general. In addition, we argue that any real movement cannot exist without a topological environment. In this sense the inclusion of such a map makes SMM more realistic.

Second, the trace data we use are coarse-grained while the synthetic movement data generated by SMM are fine-grained (as typically required by any simulation study of a mobile system). This inevitably requires certain interpolation procedure. We accomplish this by taking advantage of topology information provided by the map and by adopting simple heuristics based on distance. The validity of this approach is verified by comparing the data generated by SMM and real measurements of pedestrians on campus, as we show in the next section.

Figure 1 illustrates the methodology underlying the generation of SMM. As shown, the SMM generating system accepts two inputs: wireless trace data and a map. The

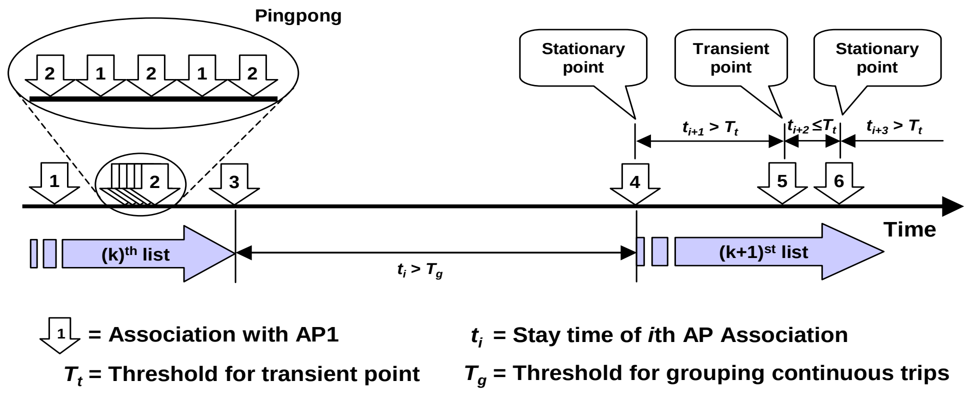


Figure 2: Definitions and concepts in our initial processes in Section 3.1.

map provides graph information in the form of vertices and segments, and the coordinates of APs, buildings, and intersections, as detailed in Section 3.2. Our system produces two outputs: a set of transition probabilities (Section 3.5) and user densities (Section 3.6). These two outputs also constitute the statistical mobility model. Specifically, the user density is used by the model to generate initial positions of mobile users in the network along with their destinations. The transition probabilities are used to generate in a sequential way the movement of an individual user (in the form of a randomly selected path between her origin and destination). These are discussed in more detail in Section 3.7. When we use these two components to generate actual movement, we also need to adopt certain timing/speed aspect that can specify a user's speed (e.g., randomly chosen). Under our method this aspect is added as an external component independent of the statistics obtained from the trace data. More is discussed in Section 3.7.

3.1 Filtering Trace Data

The trace data we used for this study were collected over a campus-wide wireless network at Dartmouth College [1]. The traces include observations from syslog, SNMP, and tcpdump. Among these, we used syslog to build SMM. One could also imagine using SNMP data, but we did not do so. Our traces contain the log data transmitted from each AP—Cisco 340 and 350 running VxWorks and Cisco IOS. It records times at which a user's wireless card is associated, authenticated, roamed, re-associated, disassociated, or deauthenticated. Each line of log data lists the AP name, MAC address of the wireless card, and the type of message transmitted. There were some APs owned by Dartmouth but taken off-campus captured in the original trace, but these were removed after the fact.

These traces were filtered to generate a chronological list of "trips" taken by each user. Each trip consists of an origin, a destination and any possible intermediate locations, denoted by an AP or building. For simplicity, we coalesce different APs in the same building to one location.

The filtering process consists of three steps: separating discontinuous trips, smoothing out high frequency pingpong phenomena, and identifying a trip with stationary/transient points. Figure 2 shows definitions and concepts that were used in these steps. We explain each step below.

3.1.1 Initial Processing

The first step of initial processing is to separate the syslog data, which contain information of all users, into traces for individual users. There are primarily two types of users present in these traces, the laptop users and the VoIP phone users, identified by the OS type of mobile device in the trace data. It turns out that the VoIP trace is more fine-grained than the laptop trace in that the former contains more association logs with intermediate APs whereas the latter rarely contains ones. The reason is because a VoIP phone user is more likely to keep his wireless device on while moving, and therefore to register with all APs he passes on his way. By contrast, a laptop user typically turns the laptop off when moving, and therefore does not register with APs he passes through. Our methodology applies to both types of users. Therefore throughout our discussion we will use the term *trace data* to refer to both types unless specified otherwise.

The result of the above is a list of successive APs accessed by each user. The second step of the initial processing is to segment such a list into separate and shorter lists based on the timestamps. Specifically, if two successive AP logs (say AP 1 and AP 2) are time stamped more than T_g hours apart, T_g being some threshold parameter which is tunable, then we segment the list into two, with the first one ending at AP 1 and the second one starting at AP 2 (see Figure 2). By repeating this process we obtain shorter lists of successive AP accesses that have occurred closer in time. The rationale behind this step is that two successive accesses with a long time lag in-between are likely to be unassociated with the same trip. For the rest of our discussion, an AP list refers to the shortened list resulting from the above two steps.

In the third/last step of initial processing, we compute the *average stay time* of a given AP/building as follows. Consider a sequence of AP associations denoted by the ordered list of the couple (AP_i, s_i), $i = 1, 2, \cdots$, where AP_i is the ith AP association that occurred at time s_i, and $s_i \leq s_j$ for $i < j$. We will define the *stay time* of AP_i, denoted by t_i, as

$$t_i = s_{i+1} - s_i \ ,$$

i.e., the stay time at an AP is the time lag between this AP and the next one. The average stay time at an AP, say AP 1, denoted by $\bar{t}_{AP1}$, is the total stay time at AP 1 divided by the number of occurrences of AP 1 in the AP list. This average is computed for each user separately, reflecting on average how much time a particular user spent at different buildings. The notion of average stay time is used for smoothing out the pingpong phenomenon and for identifying a trip, discussed in the next two subsections, respectively.

3.1.2 Smoothing out the pingpong phenomenon

The so-called "pingpong" phenomenon was originally reported by [10, 24]. It happens when a user is within the coverage of two or more APs. The wireless channel fluctuation causes the estimated signal strength from these APs to fluctuate as well. For example, the log data may show that user A was associated with AP 1 at some time s, and associated with AP 2 located 100m away after 1 second. Another second later, the log says that the user was associated back with AP 1. Considering practicality, it is unlikely that the user physically moved from one AP to the other within seconds. A more plausible explanation is that the user was somewhere within the range of both APs, and the rapid switching between APs is due to the fact that the wireless device was trying to associate with a stronger signal. When the pingpong phenomenon occurs, there may be many entries in the log that do not necessarily suggest significant movement on the user's part. However, it is not always clear when such entries should be recognized as a pingpong event. For instance, if the same sequence of AP associations were observed, but with time gaps on the order of minutes instead of seconds, it is hard to tell whether this is a pingpong event or not (and thus hard to tell if real movement had occurred). It is clearly possible for a VoIP phone user to visit a neighboring building and come back in several minutes.

While there is no perfect solution to this problem due to incomplete information and the ambiguity in its interpretation, we can reduce and smooth out high frequency pingpong switching by applying a moving average filter to the logs as follows.

Again consider a sequence of AP associations (AP_i, s_i) and the corresponding stay times $t_i = s_{i+1} - s_i$, $i = 1, 2, \cdots$. Then for the ith entry on the list that occurred at time s_i, we can estimate the "average location" of the user by using his registered locations in the past (within a time window W), weighted by their respective stay times:

$$(\bar{x}(s_i), \bar{y}(s_i)) = \frac{\sum_{k=0}^{n_i(W)} t_{i-k} \cdot (x_{i-k}, y_{i-k})}{\sum_{k=0}^{n_i(W)} t_{i-k}}\ , \qquad (1)$$

where $(x_i,\ y_i)$ is the x and y coordinates of the ith AP on the list, $n_i(W)$ is the number of AP associations within the window W that ends at s_i. In other words, $s_{i-n_i(W)-1} < s_i - W < s_{i-n_i(W)}$. This average is calculated by giving more weight to buildings with longer stay times.

This average computation is done for every AP entry on the list, i.e., producing the couple $(\bar{x}(s_i), \bar{y}(s_i))$ for the AP_i, $i = 1, 2, \cdots$. We then replace the ith entry AP_i with an AP whose location is closest to the computed average $(\bar{x}(s_i), \bar{y}(s_i))$, denoted by $\widehat{AP_i}$. At the end of this step, we group consecutive appearances of the same AP into one log entry with the stay time being the sum of all.

Figure 3 shows an illustration of applying this smoothing method, where Figure 3(a) gives the original list of AP_i, Figure 3(b) gives the result of averaging, the list of $\widehat{AP_i}$, and Figure 3(c) shows the result after merging a sequence of the same APs. As shown, an indication of the pingpong phenomena between APs *Bldg1* and *Bldg2* can be observed in Figure 3(a). By averaging over a window of $W = 1800$ seconds, the occurrences of *Bldg2* is removed. Similarly, between *Bldg3* and *Bldg4*, the former is averaged out.

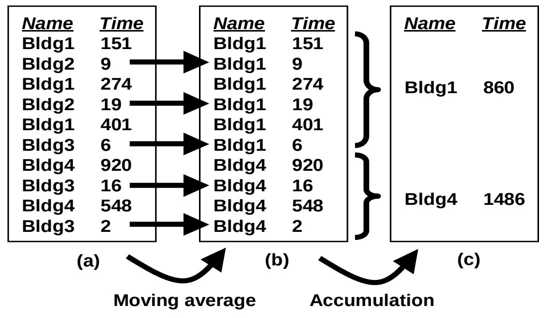


Figure 3: An illustration of applying the moving average filter weighted by stay times with $W = 1800$ seconds.

The choice of the moving window size W has a number of implications. In general, the larger the window size is, the more AP entries with small stay times we filter out. On one hand, large W can provide a better guarantee of the elimination of pingpong events. On the other hand, large W also increases the likelihood of accidentally removing AP entries that are part of some actual movement. That is, a large value of W leads to an increased miss rate, whereas a small value of W leads to an increased false alarm rate. For example, a user that moves from AP 1 to AP 3 via AP 2 registers with all three APs. Since he is only passing by AP 2, the stay time at AP 2 is very short. If W is set too large, due to the fact that the average is weighted by stay times, AP 2 could be removed from the AP list (see Figure 3(c)). This is undesirable, since the association with AP 2 could provide us with valuable information as to which route the user takes to travel from AP 1 to AP 3. This is particularly relevant to VoIP traces, because such intermediate points are more often available.

To address this problem we apply different W values to different APs/buildings based on their average stay time ($\bar{t}_{AP}$) calculated during the initial processing. The idea here is that the average stay time is a measure of how likely a particular building is used by a particular user as a passing-through building or a building where he spends significant time (e.g., where his office/classroom is). By comparing a user's average stay time at a building with a pre-set threshold T_t, we can decide whether the building falls into the first category (i.e., $\bar{t}_{AP} \leq T_t$), or the second category (i.e., $\bar{t}_{AP} > T_t$). For the first type of buildings, W is set to a smaller value (e.g., 10 minutes as used in our experiment), and for the second type, W larger (e.g., 30 minutes as used in our experiment).

The end result of this filtering step is a shortened and consolidated AP list with not only short-lived redundant AP entries removed but also loss of valid intermediate points minimized.

As an aside, one could imagine using pingponging to constrain a user's location to the intersection of their coverage areas. Unfortunately, Dartmouth's deployment density is high enough that this is not worth the increased complexity in filtering logs.

3.1.3 Identifying a trip and its origin/destination

The previous smoothing process filters out short-lived entries on the AP list. From the resulting list of AP associations, we can infer the trips a user has taken. We describe this process next.

In general the origin, destination, and intermediate points along a trip may be inferred by examining the stay times at these locations. If the stay time is significant, we may regard the location as the destination of the previous trip and/or the origin of the next trip. Specifically, for each AP entry on the list, we will examine both its stay time of that entry and its average stay time calculated earlier. If *either* quantity is larger than the threshold T_t (same as the threshold introduced for determining the window size W), this AP entry is marked as a *stationary point* for the user as shown in Figure 2. If *both* quantities fall below or equal to T_t, then this AP entry is marked as a *transient point* for the user.

Once we complete this process for every AP entry on the list, the entries between two consecutive stationary points are recognized as a "trip", and the two stationary points are recognized as the origin and destination of that trip, respectively. Any possible entries between the origin and destination are recognized as intermediate points/locations. Thus we have turned the original AP list into multiple lists, one per trip.

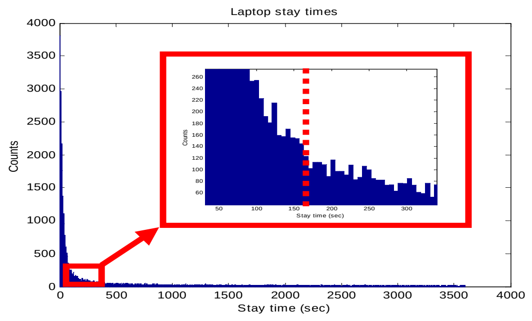
of course, this inference method is by no means unique or perfect. In particular, anything that occurred during the gap between any two consecutive time instants, if not recorded, is not known. This is the inherent limitation of coarse-grained data traces. While VoIP trace data are more fine-grained than laptop trace data, it also has a significant amount of gaps. Nevertheless, as we show in Section 4, combined with heuristics in reasoning about the routes a user takes, this method generates mobility patterns that are quite realistic.

In our processing we set $T_t = 3$ minutes. This allows enough time in general for a user to pass by a building on foot. This quantity was selected also because the histogram of the stay times at buildings, as shown in Figure 4, indicates that the decrease in the distributions for both laptop and VoIP phone users slows down noticeably at around 3 minutes, which may be viewed as the division between the more frequent transient points and the less frequent stationary points.

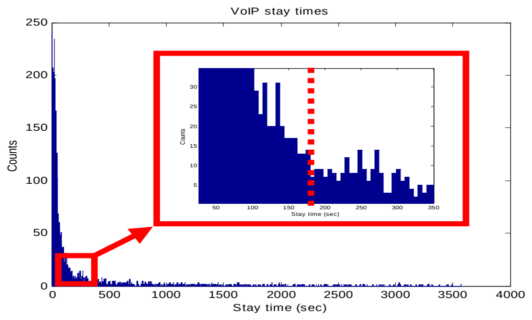
3.2 Converting a map to a graph

The previous trace filtering step gives us the origin, destination and intermediate APs that a user accesses during a trip. The next step is to infer the route/path the user takes on that trip. In order to do this, we need to have a proper representation of the geographical environment, in which each route is uniquely defined and identified. In this subsection, we illustrate the procedure we take to convert a conventional campus map to a graph that consists of vertices and links (or segments) for our purpose. Figure 5 shows the steps of this conversion process.

The trace data were properly anonymized such that the identity of each AP is not known (i.e., the building in which they are located). However, the *relative* coordinates of them are available. At the same time, we can also find the other relative coordinates of buildings from the conventional campus map. This allows us to match all APs that have been actively used and recorded in the trace data on the campus map, to the building they reside in with a numerical index. As mentioned earlier, for simplicity we consider all APs in a single building as one and the same. Thus each building has only one number representing all APs in the building as shown in Figure 5(a). These buildings will be used as origins and destinations for the SMM to generate user movement.



(a) Laptop stay times (sec).



(b) VoIP stay times (sec).

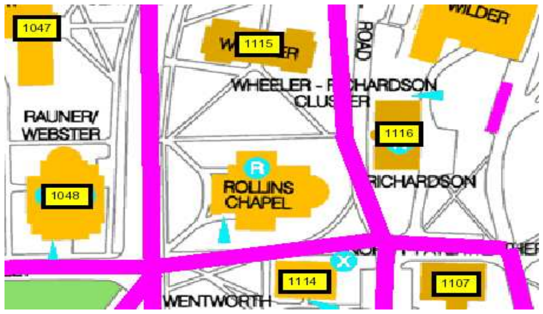
Figure 4: Histograms of laptop and VoIP stay times at buildings within 1 hour from the trace data in Section 4. Dotted lines on the magnified graphs show the points from which the decrease seems to start to slow down noticeably.

In mapping out the roads/paths connecting these buildings, again for simplicity we will limit our selection to only the major roads plotted on the map as shown in Figure 5(a). All roads are approximated and represented by piecewise linear segments (e.g., a curve consists of multiple line segments).

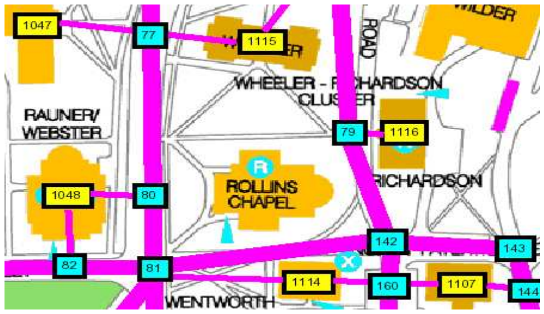
The buildings are then connected to the adjacent major roads via entrances and exits as illustrated in Figure 5(b). Specifically, buildings are either connected to the closest major roads via a pathway perpendicular to the major road, or connected to the closest intersection. In reality, a building may have many entrances and exits at various locations around it. In our conversion process, we considered only the main entrances that most people use. Minor buildings are assumed to be accessible from the most adjacent major road for simplicity.

Once all roads and connections between roads and buildings are included on the map, intersections are enumerated as shown in Figure 5(b). The intersections are uniquely identified, because they are used as the decision points for the SMM to generate random routes as we show in the next subsections. When a curved road is approximated by a sequence of line segments, the connecting point of two adjacent segments is also considered as an intersection. By doing so we convert the original roads/paths into a set of links.

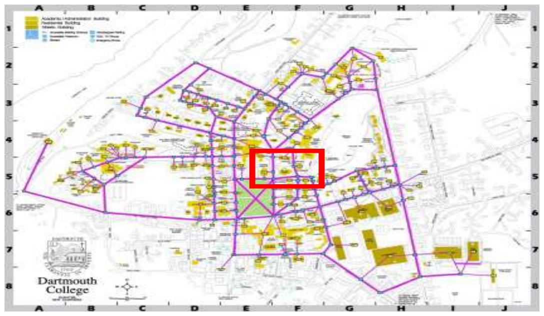
Figure 5(c) is the final map produced after the above preliminary processes. A small rectangle in the middle of Figure 5(c) is the area shown in Figures 5(a) and (b) above. Figure 5(d) is a final graph representation with vertices (i.e., squares as buildings, and dots as intersections) and links (i.e., roads).



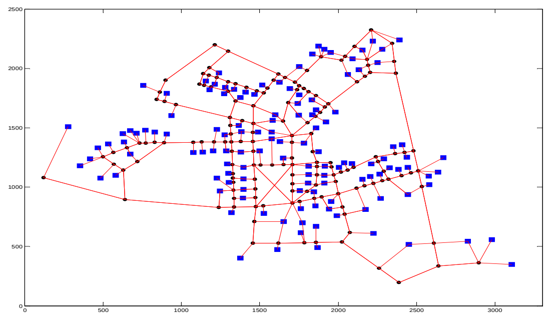
(a) Step 1: Enumerating buildings on the map as vertices and identifying major routes as segments.



(b) Step 2: Connecting buildings to adjacent major routes and enumerating intersections as additional vertices.



(c) Final map: A rectangle on the map is the area shown in Steps 1 and 2.



(d) Graph with vertices and segments.

Figure 5: Converting a map to a graph

3.3 Route candidates generator

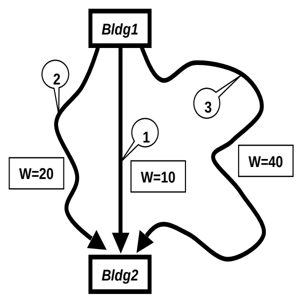
We now have a graph consisting of vertices and links. With the origin/destination/intermediate points (shown as vertices on the graph) of a trip given by Section 3.1, we are ready to infer the routes a user takes on this graph. In this section, we focus on generating the likely routes taken by a user, i.e., route candidates, on this graph, and in the next section we derive the likelihood of a route candidate being used.

In generating route candidates, we observe that the shortest route between an origin and destination is often the preferred choice. However, this is not an absolute rule. What constitutes the shortest route highly depends on the metrics people use. For example, the shortest *distance* route is usually different from the shortest *time* route (e.g., MSN Maps & Directions [2]). Other potential metrics include number of turns, popularity, safety index, crime rate, scenic rank, and so on. User preference also varies from person to person, and from time to time even for the same person. For example, a person may choose the quickest route to a destination, while taking a more scenic route on the way back. He might also prefer a secluded road for privacy during the day, while favoring a safer and more popular road at night. This motivates us to have a *set* of route candidates to select from rather than a single shortest route defined in a specific way.

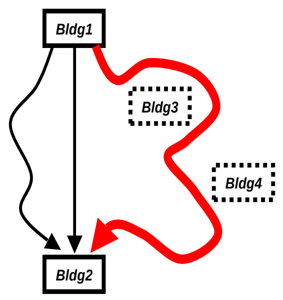
In our experiment, we have used distance as a metric to measure a route. This is primarily because distance is easier to compute and manage given the coordinates, compared to metrics like time or popularity (e.g., user density). In particular, time is not a good metric because it requires speed information which we do not have. It is also difficult to measure a user's departure time in the trace data, because in most cases a user simply shuts off his machine without dissociation with AP before leaving. Popularity is not a good metric either, not only because it is hard to measure or collect, but also because using popularity as the only metric may distort route search results. For example, all of the searched and selected routes from a building to its neighbor could be several times longer detours, not the obvious straight route between them, if the measured popularity between two buildings is extremely low. So we stick to distance as a metric for route searching.

Once the metric is determined, we can search for a set of routes available between the origin and destination by using an algorithm of finding N shortest paths (see for example [16, 17]). This algorithm finds the shortest path up to the Nth shortest one. We used $N = 10$, 30, and 50 in our experiment. Comparison will be given in Section 4.

Even when there are a number of geographically available routes between two buildings, not all routes can be considered because the kth shortest path for some large k tends to result in a path that winds through buildings which does not appear to be realistic. Therefore we exclude the routes that are practically unlikely from the set generated by the route search algorithm by setting a certain threshold C on the route distance compared to the shortest path. The idea is that it is reasonable to assume that most people are unlikely to take a detour instead of shorter ones if that detour exceed in distance by a certain amount. For example, $C = 2\times$ means that the distance of route candidates does not exceed more than $2 \times$ the shortest distance for each pair of origin and destination. We used $C = 2\times$, $3\times$, $4\times$, and no limit in our experiment, as will be shown in Section 4.



(a) 3 route candidates between *Bldg1* and *Bldg2* with weights of 10, 20, 40, respectively.



(b) Route 3 is selected if transient points are *Bldg3* and *Bldg4*.

Figure 6: Selecting the most probable route between the origin and destination.

3.4 Selecting and counting routes

With the route candidates generated in the previous subsection, we now proceed to determine the likelihood of each route being used. We do this by counting the frequency of usage of each route throughout the trace data, i.e., the AP lists resulting from the filtering process described in Section 3.1, over all trips taken by a user for all users.

Specifically, consider a trip with two consecutive stationary points *Bldg1* and *Bldg2*. Suppose there are three possible routes between *Bldg1* and *Bldg2* generated by the route candidate generator described in Section 3.3. Each route is labeled with a *weight* indicating its preference as shown in Figure 6(a). Since we use distance as a metric, the weight of each route is simply the total distance of that route. The inference follows that the user has visited *Bldg1* and moved to *Bldg2* through one of these three routes. Thus this observation will be added to the frequency of one or more of these routes. For example, ideally if we knew that the user used route 1, then the usage frequency of route 1 would be incremented by 1 while the frequencies of routes 2 and 3 would remain unchanged, and the same process would be repeated over all trips taken by all the users.

Since we do not know exactly which route is taken on a particular trip, we consider the following two possible cases in this incrementing/counting process. In the first case, there are no transient points between the two consecutive stationary points *Bldg1* and *Bldg2* in the trace data. Thus there is no indication as to which route more likely has been taken from the trace data. In this case we will assume that the frequency of taking a route is inversely related to its weight/distance, and simply distribute this single observation among all three routes in different proportions as follows. Route i gets an increment of Δ_i given by

$$\Delta_i = \frac{\frac{1}{w_i}}{\sum_{k=1}^{N} \frac{1}{w_k}} \tag{2}$$

where w_i is the weight of route i, and N is the total number of route candidates generated for the origin-destination pair. In other words, we increment the frequency of a shorter route by more, assuming that a shorter route is more preferred than a longer one. For example, in Figure 6(a), routes 1, 2, and 3 have weights 10, 20, and 40, respectively. Their frequencies are incremented by $\frac{\frac{1}{10}}{\frac{1}{10}+\frac{1}{20}+\frac{1}{40}} = 0.57$, $\frac{\frac{1}{20}}{\frac{1}{10}+\frac{1}{20}+\frac{1}{40}} = 0.29$, and $\frac{\frac{1}{40}}{\frac{1}{10}+\frac{1}{20}+\frac{1}{40}} = 0.14$, respectively, for this single observation of the pair of $Bldg1$ and $Blag2$ in the trace data. Note that $\sum_{i=1}^{N} \Delta_i = 1$, i.e., the increments over all routes add up to 1 over a single observation/trip.

In the second case, transient points are registered between the two consecutive stationary points as shown in Figure 6(b). In this case we may use this extra information to infer which route likely has been taken. Suppose two transient points *Bldg3* and *Bldg4* were observed between *Bldg1* and *Bldg2* in the trace data with their respective locations as shown in Figure 6(b). It follows that route 3 is likely to be the route taken on this trip from *Bldg1* to *Bldg2*, and the frequency of route 3 is incremented by 1. To translate the occurrences of these transient points into a particular route, we utilize the distance information. Specifically, we compute the Euclidean distance between each transient point and each of the route candidates (i.e., the distance of the connections between buildings identified as these transient points and their closest main roads). We then select as the most likely route the route whose sum of distances to each of the transient points is minimum. In Figure 6(b) route 3 gives the minimum. The usage frequency of route 3 is then incremented by 1 over this single observation/trip of the pair of *Bldg1* and *Bldg2*.

The above counting/incrementing process is repeated for all routes over the entire trace data, i.e., all trips taken by all users. The resulting usage frequencies are then used to derive transition probabilities and user density, discussed in the next two subsections, respectively.

3.5 Transition Probabilities

Technically the route frequency statistics produced by the counting process described in the previous section could be used directly to derive a probabilistic mobility model. Imagine a set of users randomly generated over the vertices on the graph (user density is discussed in Section 3.6) in simulating a certain mobile system. Thus each route is associated with an initial origin. For each of these users, we can select a route (including its destination and the specific path between the origin and destination) with a probability consistent with the route frequencies statistics, and repeat the same process.

Alternatively, we can also model the user mobility as a Markov chain and turn the route frequency statistics into a set of *transition probabilities* associated with each vertex (including all building locations and intersections) on the graph. With this approach, once we have all the transition probabilities we no longer need to store the routes and their frequencies. In using the resulting mobility model, we simply decide for each user moving on the graph his next intersection given his current and past positions according to the transition probabilities. With the right transition probabilities this approach would generate statistically similar routes and movement patterns to that generated by the first approach. The advantage of using this approach is that transition probabilities provide more direct information than the first approach. Note that turning route frequencies into transition probabilities at intersections directly tells us user mobility information like ratio of pedestrians in each direction (which we use for evaluation in Section 4) at each intersection. This is not directly available from route frequencies, even though the same information can be obtained via methods like computing the transition probabilities.

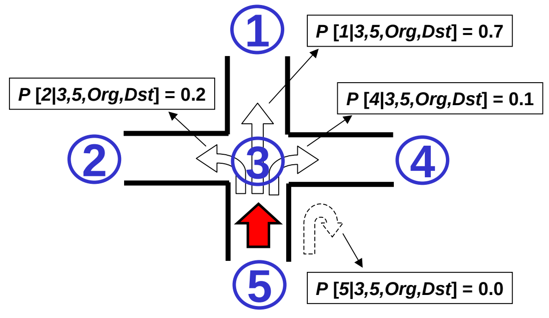


Figure 7: Transition probabilities at the intersection in a form of P [next intersection | current and previous intersections, origin, destination]. A decision should be made at intersection 3 when a user arrives from intersection 5 during the trip from the origin *Org* to the destination *Dst*.

The SMM is built using the second approach, and the model itself is essentially a second-order Markov chain, given by a set of transition probabilities and a stationary probability distribution (user density). Specifically, a transition probability for this second-order Markov chain is defined by the following conditional probability:

$$Prob[\text{next} \mid \text{current, previous, origin, destination}]$$

where all terms refer to an intersection or a building location. Within the context of Markov chain modeling, the position of a user (the vertex where he is) will also be referred to as the *state* of the user [6, 12]. As can be seen from this definition, a user's next state is determined by his current and previous states (thus second-order) in addition to his origin and destination. An example is shown in Figure 7.

Using conditional probabilities P [**n**ext | **c**urrent, **p**revious, **o**rigin, **d**estination], if one of the routes between origin O and destination D is $O = v_1 \longrightarrow v_2 \longrightarrow \cdots \longrightarrow v_n = D$, then the probability of the route being taken among all routes between the $O - D$ pair is simply given by

$$\begin{aligned} &P[O = v_1, \cdots, v_n = D|O, D] \\ &\quad = \quad P[v_2, \cdots, v_n|O, D] \\ &\quad = \quad P[v_2|O, D] \prod_{i=3}^{n} P[v_i|v_{i-1}, v_{i-2}, O, D]. \end{aligned} \tag{3}$$

The transition probability P [**n** | **c**, **p**, **o**, **d**] can be obtained from the route usage frequencies, by treating each route as a sequence of decisions made by the user. The transition probability at a particular vertex/intersection is then computed as the ratio between the number of decisions made in each direction (i.e., next vertex) and the total number of decisions made at this vertex given by

$$\begin{aligned} &\widehat{P}[\,\mathbf{n} \mid \mathbf{c},\ \mathbf{p},\ \mathbf{o},\ \mathbf{d}\,] \\ &\quad = \frac{\text{Number of decision } \mathbf{n} \text{ at } \mathbf{c}\text{, given } \mathbf{p}, \mathbf{o}, \mathbf{d}}{\text{Number of all decisions at } \mathbf{c}\text{, given } \mathbf{p}, \mathbf{o}, \mathbf{d}}. \end{aligned} \tag{4}$$

For example, in Figure 7, suppose from the route frequencies we count that the decision of going to vertex 1 was made 70 times at vertex 3, given the previous vertex was 5 while moving from origin *Org* to destination *Dst*. Suppose further that the decisions of going to vertices 2 and 4 were made 20 and 10 times, respectively, and that nobody turned back to vertex 5 from vertex 3. Then the total number of decisions made at vertex 3, given the previous vertex 5 with the origin Org and the destination Dst, is $70 + 20 + 10 = 100$. Thus the transition probabilities are $P\ [1 \mid 3, 5, Org, Dst] = 70/100 = 0.7$, $P\ [2 \mid 3, 5, Org, Dst] = 20/100 = 0.2$, $P\ [4 \mid 3, 5, Org, Dst] = 10/100 = 0.1$, and $P\ [5 \mid 3, 5, Org, Dst] = 0/100 = 0.0$.

Note that the transition probability in Equation (4) is only an estimate (typically denoted by $\widehat{P}[\cdot]$) of the true value $P[\cdot]$ which is unknown. This is because these transition probabilities are obtained through empirical means by counting the relative frequencies. They approximate the true values well when the sample size is sufficiently large as we assume in this study. The notation $P[\cdot]$ is used instead of $\widehat{P}[\cdot]$ for simplicity without causing ambiguity.

3.6 User density

The transition probabilities we obtained above describe how users move around on a graph during a simulation. One more remaining question is the determination of initial distribution and deployment of these users. This corresponds to the initial state distribution of the Markov model we developed in the previous subsection, and is also referred to as the user density.

From the route frequency statistics collected in Section 3.4, we can easily determine the initial distribution of origins by counting the frequency of visits to buildings when they occur as stationary points in the trace data. These statistics give the likelihood of selecting a certain building as an origin. Although we can compute both user densities in the buildings and on the roads, the user density in the buildings is more appropriate for deploying mobile users with the transition probabilities, because our transition probabilities are conditioned on the stationary points as an origin and a destination.

Similarly, we can also compute the destination distribution for each origin building. This is done by counting the relative frequency of each building that appears as a destination (the second stationary point of a pair in the trace) for a given origin (the first stationary point of a pair). This distribution is used to determine the destination of a user's trip given his origin of that trip in the next subsection.

3.7 Implementation of SMM

Given transition probabilities and user density, we know how to determine where users start from and how they move around. These two elements essentially constitute a statistical mobility model.

Following the approach outline in this section, we have built an SMM scenario generator by modifying `setdest`, a utility program in ns-2 [3] that generates movement scenarios for ad hoc network simulations. The modified `setdest`, named `smm`, accepts six input files: (1) graph information with vertices and links, (2) their coordinates, (3) transition probabilities, (4) user density over the buildings, (5) destination distribution over the buildings, and (6) the average stay times in each building. Note that the graph information and coordinates are obtained in Section 3.2, transition probabilities in Section 3.5, user density and destination distribution in Section 3.6, and average stay times in Section 3.1. The route candidates generator and the rest of SMM generating system in Figure 1 are written in C++ and Perl, respectively.

SMM works as follows. First, the user deployment process randomly picks a building as an origin from the user density, and selects another building as the destination from the

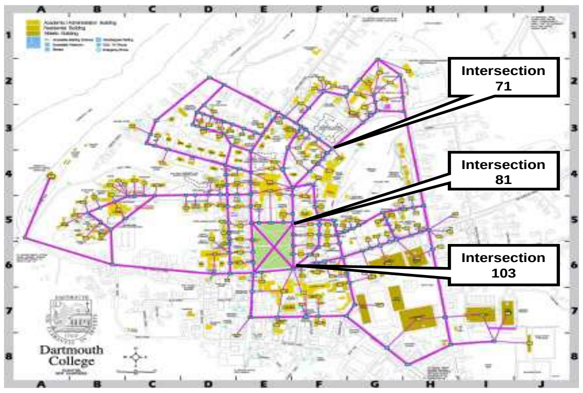


Figure 8: Intersections 71, 81, and 103 on the campus map for the evaluation.

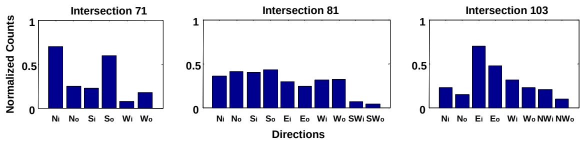
destination distribution. This process is repeated for each user in the system. Next, the scenario generating process specifies the user movement on the graph according to the transition probabilities. Once a user arrives at the destination, he stays there first for the average stay time of the building. Then he selects the next building from the destination distribution of the current building (i.e., origin for the next trip). The same process is then repeated. A user's speed is selected at random whenever he arrives at a vertex and makes a decision on the next direction. Minimum and maximum speeds are specified beforehand. For example, the speed range from 0.5 to 1.5 m/s would be reasonable if people are assumed to be on foot. The initial speed distribution is given by the stationary distribution to avoid the transient period in the beginning of the simulation [26].

4. EVALUATION OF SMM

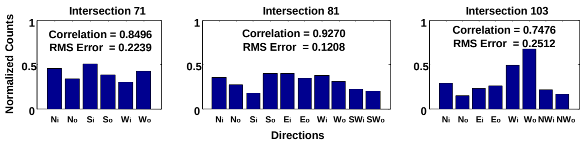
To evaluate our methodology, we manually counted the number of pedestrians passing through several intersections on the campus of Dartmouth College, and compared these real measurements to the results (or synthetic data) produced by the SMM generated from the wireless trace-driven framework described in the previous section.

We selected three intersections on College Street passing through the center of the campus, and named them intersections 71 (Maynard Street), 81 (Wentworth Street), and 103 (Wheelock Street), respectively, following our enumeration in Section 3.2. These intersections are shown in Figure 8. At each intersection, we manually counted the number of pedestrians passing through at five or six different times for ten minutes each. This was done over 2 days on October 24 and 25, 2005. Specific observations were taken from 10 to 11 AM on Oct 25 at location 71; 1-3 PM on Oct 25 at 81; and 2-4 PM on Oct 24 & 8-10 AM on Oct 25 at 103. These times were chosen only to provide representative coverage during normal business hours. Unfortunately, we were unable to also obtain trace data for these periods, due to a change in the trace collection mechanism.

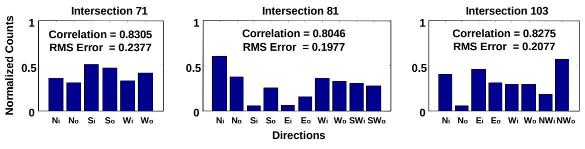
We counted people on each leg of the intersection, for both inbound and outbound directions. For example, intersection 71 has north-, south-, and west-bound legs with both inbound and outbound flows on each leg. This resulted in counting people in six different directions, denoted by N_{in}, N_{out}, S_{in}, S_{out}, W_{in}, and W_{out}, respectively. These quantities are measured simultaneously. For instance, if a person passes through the intersection from south to north,



(a) Real measurements of the number of pedestrians.



(b) SMM-generated results of the number of laptop users.



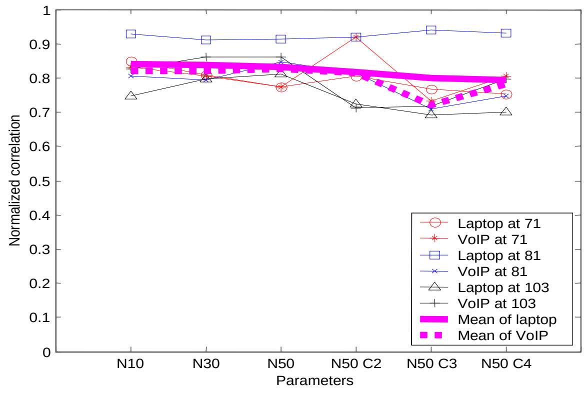
(c) SMM-generated results of the number of VoIP phone users.

Figure 9: Normalized counts of real measurements of pedestrians and SMM-generated results of laptop & VoIP phone users in each direction at intersections 71, 81, and 103. N, S, E, and W stand for north, south, east, and west legs of the intersection. Subscripts i and o denote inbound and outbound at the intersection. Parameters of max 10 routes and no cutoff were used.

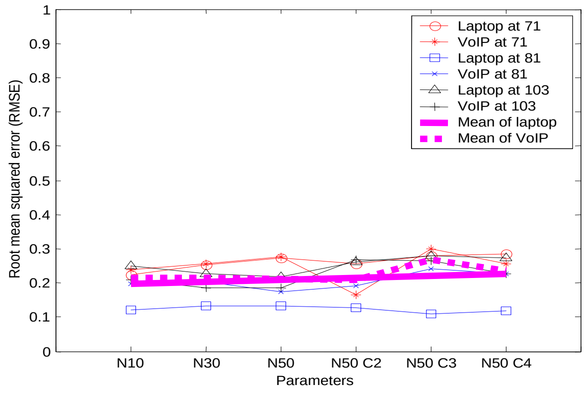
then we increment both S_{in} and N_{out}. When the traffic was light, we used hand counters. When the traffic was heavy, we videotaped the intersections and legs with a camcorder and counted afterwards by play-back. Figure 9(a) shows the real measurements of the number of pedestrians passing through intersections 71, 81, and 103, respectively. The quantities shown here have been normalized for fair comparison between different intersections. Thus, Figure 9(a) does not imply that the flows of people at three intersections are the same. The total counts per 10 minutes at intersections 71, 81, and 103 ranged from 22 to 48, 62–224, and 56–186, respectively.

Obviously, not all of these pedestrians were necessarily WiFi users. Our hypothesis was that enough Dartmouth community members use WiFi regularly to make WiFi user's movement representative of the community as a whole. As our comparisons show, this appears to be the case.

Since the wireless trace data for the same days as our measurements have not been released yet, we used the trace data collected during roughly the same academic calendar period in past years for comparison and evaluation. Specifically, our measurements are for Oct 24 (Mon) and Oct 25 (Tue), 2005, and we used the trace data collected on Mondays and Tuesdays in late October in the past three years: Oct 22/23 & 29/30, 2001, Oct 21/22 & 28/29, 2002, and Oct 20/21 & 27/28, 2003. Of these traces, we only used the data within the same time window of real measurements, plus 30 minutes at both ends of the window. These two 30-minute periods are added because for our filtering process we need to know which APs users are associated with at their origin and destination before and after passing through intersections. These traces are then filtered through the process described in the previous section, resulting in a set of



(a) Correlations of normalized counts.



(b) RMSEs of normalized counts.

Figure 10: Correlations and root mean squared errors of normalized counts of real measurements and system-generated results with various sets of parameters. N10, N30, and N50 denote max 10, 30, and 50 routes. C2, C3, and C4 stand for the cutoff thresholds of 2×, 3×, and 4× minimum distance, respectively.

transition probabilities and user densities. Figures 9(b) and (c) show the normalized counts of laptop and VoIP phone users, respectively, inferred from the above trace data with parameters of max 10 routes and no cutoff, which is denoted by $N = 10$ following our notation. To evaluate how well the SMM-generated results match the real measurements, we computed two metrics: normalized correlation and root mean squared error (RMSE), both between 0 and 1 due to the normalization. As shown in Figures 9(b) and (c), the correlation between the real measurements and the SMM-generated results at the intersections ranges from 0.7476 to 0.9270 for laptop users and from 0.8046 to 0.8305 for VoIP phone users. On average, the correlations of both types of users are greater than 0.8, implying that the real measurements and the SMM-generated results are very close. The averages of RMSEs for both types of users are around 0.2, which roughly translates into an average error of 20% or so when we use SMM-generated mobility patterns to approximate that of the real user.

Figures 10(a) and (b) show the normalized correlations and RMSEs, respectively, with various parameters including $N = 10$. We note that while there are slight variations from one parameter to another, the resulting correlation and RMSE are approximately 0.8 and 0.2, respectively, for both types of users regardless of the choice of parameters. Overall Figure 10 suggests that our methodology is quite insensitive to these parameters. Although the system with no cutoff threshold seems to show a slightly better performance than that with a cutoff threshold in Figure 10, the difference is too small to assert that one is clearly better than the other. The reason why the maximum number of routes and cutoff threshold do not affect the performance much may be found in the results in Section 5. This is because on average most laptop and VoIP phone users turned out to have taken one of the top five shortest routes among all available candidates as shown in Figures 11 and 12. Thus, maximum number of routes greater than 5 or so such as 10, 30, or 50 results in little difference in the performance. This is also true of cutoff threshold. As shown in Table 1 in Section 5, average numbers of routes with cutoff thresholds of 2×, 3×, and 4× minimum distance are much greater than 5, which again makes little difference in performance.

Another interesting observation from Figure 10 is the insensitivity of our method with respect to the granularity of the trace data. Recall that although both are coarse-grained, the VoIP data is slightly finer in granularity than laptop data. On the other hand, as we can see, the result from laptop traces is approximately the same as or slightly better than that from VoIP traces consistently. This is contrary to our expectation; one naturally expects to do better with finer-grained trace data. Unfortunately we do not have a precise answer for this. This discrepancy may simply be due to the noise in the process; recall that our results contain about 20% of error. Another plausible explanation is that the number of VoIP phone users recorded in the trace data is much smaller compared to the laptop users, and thus less representative of the real movement captured by manual measurement. Indeed, VoIP phone users in the trace data were less than 5% of the laptop users. Thus a pedestrian passing through one of those intersections we measured is more likely to be a laptop user than a VoIP phone user. This implies that the pedestrian-counting evaluation method we used is more appropriate for laptop users than for VoIP phone users.

5. CASE STUDY: TWO-MONTH TRACE

In evaluating our technique, we compared the model generated by a small trace to hand-collected observations over the same period. In this section, we apply our model generator to a longer period of trace data, taken from October 1, 2003 to November 30 of the same year. This period was chosen to be similar in character to our shorter verified period. VoIP devices (Cisco 7920 and Vocera) were newly in use, and the period is wholly contained by a Dartmouth fall term. Thus, it is reasonable to expect that the data used here contain the entire mobile activities of almost all the wireless users on campus. During the above period, the logs of 2039 laptops or PDAs and 94 VoIP phones were recorded. Although a single user might have carried both a laptop and a VoIP phone, we considered them separately. Dartmouth had over 5700 enrolled students and nearly 600 faculty and staff as of fall 2004.

5.1 Routes

As described in Section 3.3, our model generator has two parameters: the maximum set of candidate routes considered between each endpoint, and the cutoff distance used to prune candidate routes. In applying the generator to our data, we considered maximum route sets of $N = 10$, 30, and 50 for each unique origin/destination pair.

Condition	Avg Number of Routes (± Error)
max 10, no cutoff	9.98 (± 0.01)
max 30, no cutoff	29.93 (± 0.02)
max 50, no cutoff	49.89 (± 0.04)
max 50, cutoff 2×	28.43 (± 0.30)
max 50, cutoff 3×	38.57 (± 0.27)
max 50, cutoff 4×	42.53 (± 0.24)

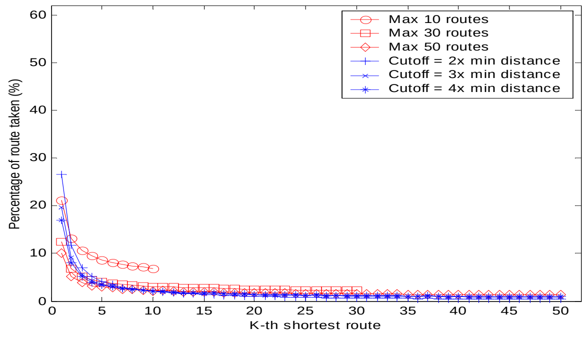
Table 1: Average number of the routes generated with various parameters. *Max* means the maximum *number* of routes allowed per pair of origin and destination. *Cutoff* means the maximum *distance* of routes allowed per pair of origin and destination. For example, cutoff 2× means the maximum distance is 2× minimum distance. Error indicates the confidence interval of 95%.

Without cutoff constraints, the route generator can find the maximum number of candidates nearly every time, no matter how close the origin and destination are; this is shown in Table 1. But, this would make no sense in the real world if the origin and destination are neighbors, because people are not likely to use 50 different routes to visit the building next door. If we apply the cutoff threshold as shown in Table 1, we reduce the number of clearly senseless route candidates. However, there are still more than 20 route candidates per origin and destination, even though we apply the cutoff $C = 2\times$ minimum distance; farther trips tend to generate more candidate routes.

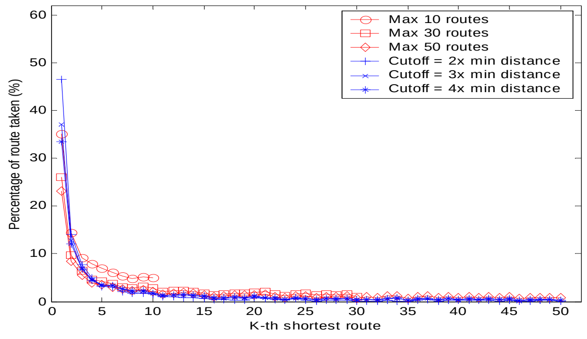
Among more than 20 candidates for each origin and destination, which route is most likely to have been taken by laptop and VoIP phone users? There are two cases to consider: with and without transient points between the origin and destination in the trace data. Recall that selecting with transient points picks only one route as the most probable route, while selecting without them coalesces a set of candidates inversely proportional to their total route weight as shown in Figure 6.

First, we made no distinction between the two cases. Figure 11 shows the percentage of each route taken by laptop and VoIP phone users, counting the most probable route in both cases. Without this distinction, most of the laptop and VoIP phone users in our system have taken a route among the top five shortest route candidates, regardless of the maximum number of route candidates N and the cutoff threshold C. VoIP phone users seem to have preferred the shortest route more than laptop users in Figure 11 in that VoIP trace data are more fine-grained and thus more likely to have transient points between the origin and destination than laptop trace data. Thus, VoIP trace data enables us to pick a single most probable route rather than consider a variety of candidates.

If we only look at the cases with transient points between the origin and destination, the tendency to select the shortest route becomes more pronounced, as shown in Figure 12. No matter what N and C are, the shortest route is taken from nearly 40% to 60% of the time. Furthermore, the top five shortest routes are nearly always taken, and the rest are negligible. This result explains why the performance of our system is not affected significantly by the parameters N and C, as long as the average number of routes per origin and destination is much greater than five or so, which is the case here.

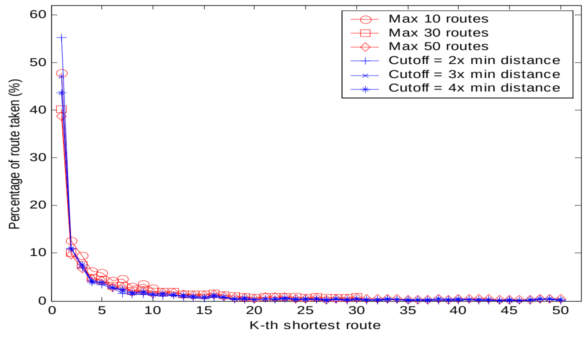


(a) Laptop users.

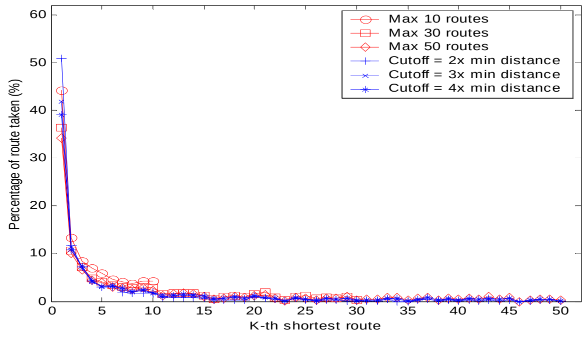


(b) VoIP phone users.

Figure 11: Percentage of the Kth shortest route taken by laptop and VoIP phone users with various parameters.



(a) Laptop users.

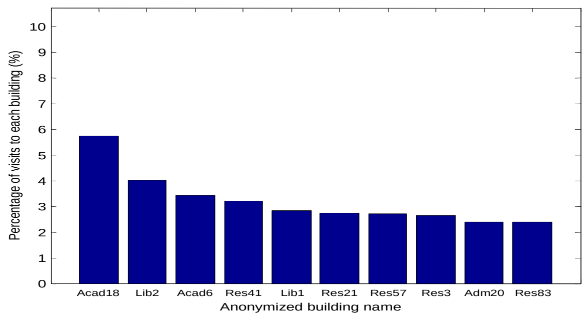


(b) VoIP phone users.

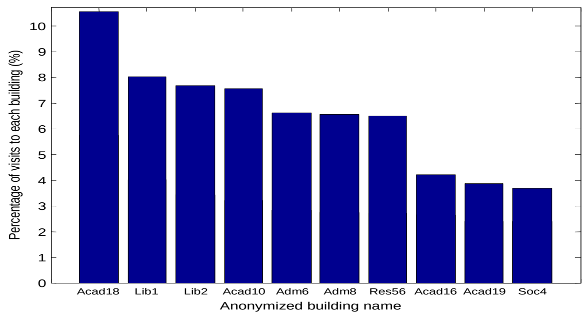
Figure 12: Percentage of the Kth shortest route taken by the laptop and VoIP phone users, only considering routes with transient points.

# of Trans. Pnts	Laptop Data	VoIP Data
0	182911 (92.3%)	4277 (66.7%)
1	8142 (4.1%)	1024 (16.0%)
2	6074 (3.1%)	437 (6.8%)
3	638 (0.3%)	274 (4.2%)
4	234 (1.2%)	174 (2.7%)
5	101 (0.5%)	90 (1.4%)
6+	116 (0.6%)	139 (2.2%)

Table 2: Percentage of the routes observed in the laptop and VoIP trace data with various number of transient points between the origin and destination.



(a) Laptop users.



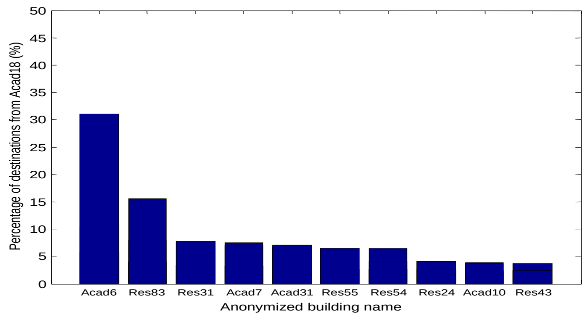
(b) VoIP phone users.

Figure 13: Top 10 buildings visited by laptop and VoIP phone users. *Res*, *Lib*, *Adm*, *Acad*, and *Soc* stand for residential, library, administration, academic, and social buildings, respectively.

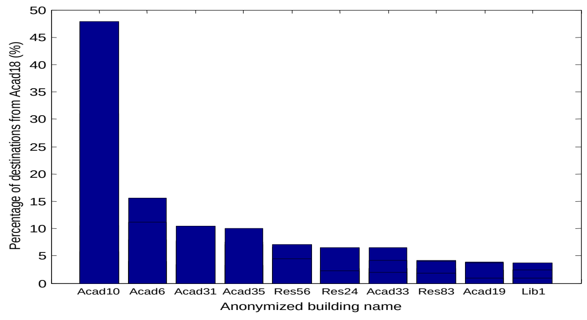
Although the cases with transients points between the origin and destination can help us pick the most probable route more clearly, we cannot ignore the cases without transient points between them, because a majority of the trips observed in the trace data correspond to nomadic laptop users—they do not have any transient points between the origin and destination; the breakdown of trips with and without transient points is shown in Table 2. Laptop trips with transient points are less than 10% of the total, while the VoIP traces hold more than 30%.

5.2 Origin and destination

Stationary points form the basis of origins and destinations in our model generator. Figure 13 shows top ten popular buildings visited by laptop and VoIP phone users during the trace collection period. Recall that buildings are categorized into several groups such as residential, library, administration, academic, and social buildings. According to the categorization, there is an interesting difference of popular buildings between laptop and VoIP phone users. Among top ten popular buildings, laptop users ranked five residential buildings, while VoIP phone users preferred somewhere else such as academic or administration buildings. This may imply that people use their laptop mostly at home and their VoIP phone at work (e.g., class or office). However, the most popular building for both laptop and VoIP phone users turned out to be an academic building. If we examine the trace data to see where laptop and VoIP phone users went from this building, shown in Figure 14, we can find the same pattern as above once again: Laptop users mostly went home for the next use (6 out of 10), whereas VoIP phone users went to other academic buildings (6 out of 10).



(a) Laptop users' destinations.



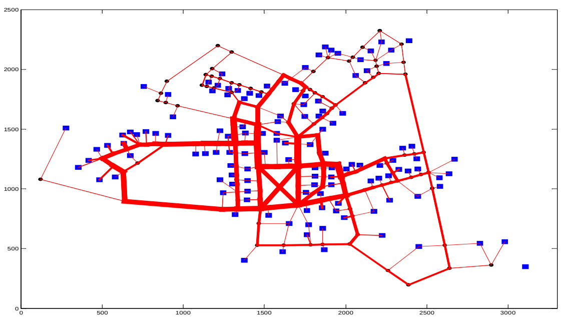
(b) VoIP phone users' destinations.

Figure 14: Laptop and VoIP phone users' top 10 destinations from *Acad18* in Figure 13.

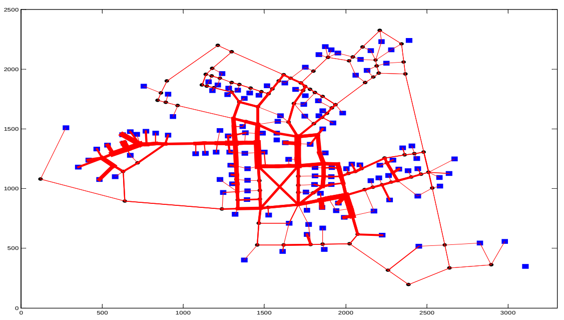
5.3 User density on pathways

We can find popular pathways by counting the frequency of routes taken by laptop and VoIP phone users, as predicted by our model. Figure 15 shows the laptop and VoIP phone user densities on the segments for two months. As in Figure 5(d), large squares and small dots represent buildings and intersections, respectively. The only difference is that the lines in Figure 5(d) indicate roads, whereas those in Figure 15 signify user density. A thicker line means a higher user density on the segment. However, unconnected dots or squares do not necessarily mean that there were no people having visited those places. It simply means that there was no log recorded in the trace data during the period. In addition, although each road may have two different user densities because pedestrians walk in two directions, we coalesce them for simplicity in Figure 15.

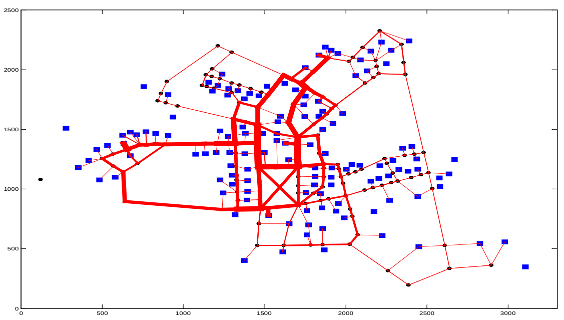
As can be expected, laptop and VoIP phone user densities on pathways are quite different. How does each set change different parameters for our generator? They are relatively insensitive to changes in N. However, if we compare the user



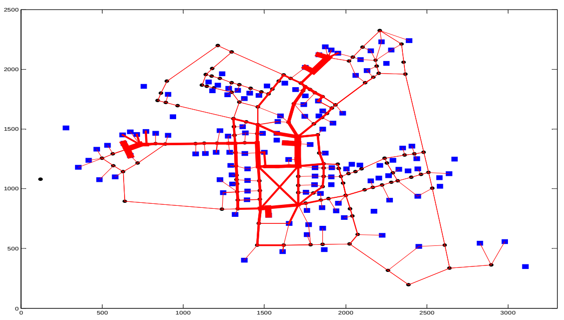
(a) Laptop user density with max $N = 30$ routes and no cutoff.



(b) Laptop user density with max $N = 50$ routes and cutoff $C = 2\times$.

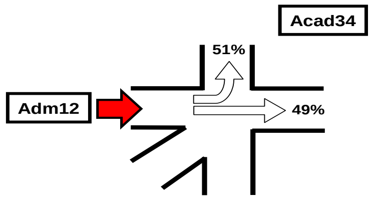


(c) VoIP phone user density with max $N = 30$ routes and no cutoff.

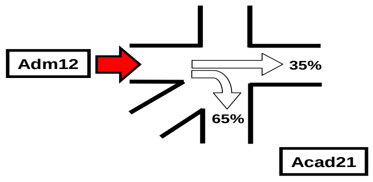


(d) VoIP phone user density with max $N = 50$ routes and cutoff $C = 2\times$.

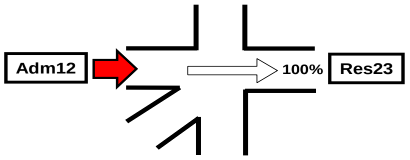
Figure 15: Laptop and VoIP phone user densities on pathways. According to Table 1, the average number of routes with $N = 30$ and that $N = 50$ and C= 2× are almost the same. But the user densities are different.



(a) From *Adm12* to *Acad34*.



(b) From *Adm12* to *Acad21*.



(c) From *Adm12* to *Res23*.

Figure 16: Transition probabilities of laptop users traveling from building *Adm12* to three different buildings through intersection 81 in Figure 8, when $N = 50$ and $C = 2\times$.

density across models generated with similar valid routes, but different parameter settings, they look somewhat different.

As an example, consider the different user densities with $N30$ and $N50, C2$, respectively. Recall that the average number of routes with $N = 30$ and that with $N = 50$ and $C = 2\times$ minimum distance are nearly 30 in Table 1. Figures 15(a) and (c) are laptop and VoIP phone user densities with $N = 30$, and Figures 15(b) and (d) are those with $N = 50$ and $C = 2\times$. It can be seen that the user densities on the road in Figures 15(a) and (c) are more or less evenly distributed and evenly thick in the adjacent area, while those in Figures 15(b) and (d) are more concentrated on the specific segments. This is because the cutoff threshold limits routes to those with shorter path lengths, resulting in more re-use of candidate route segments. Thus, the user densities in Figures 15(a) and (c) are more spread out, whereas those in Figures 15(b) and (d) are more concentrated.

5.4 Transition probabilities

We also are able to collect transition probabilities at intersections and build a transition matrix as described in Section 3.5. As an illustration, we picked an origin building about 150m west and three destination buildings about 150m northeast, 120m southeast, and 300m east from one of the three intersections that we used for evaluation in Section 4. Figure 16 shows the transition probabilities of laptop users traveling from building *Adm12* to building *Acad34*, *Acad21*, and *Res23* through intersection 81, respectively, all of which seem to be reasonable. Note that location of buildings in Figure 16 only implies their rough direction from the intersection, not their distance from it.

6. CONCLUSION

In this paper we presented a framework that generates a statistical mobility model from coarse-grained wireless trace data. Compared to other related work, this model is highly realistic in that it is defined within a geographic environment given by a map, and the movement patterns generated by this model are statistically similar to real movement collected by hand. This method infers fine-grained route information from the coarse-grained laptop and VoIP trace data with the aid of the map. We are able to obtain statistics like the user densities on the road and in the buildings, transition probabilities at each road intersection, and so on. With these statistics, we have built a statistical mobility model scenario generator for ns-2. With this method, one can easily construct a site-specific mobility model simply by plugging the site-specific trace data and the map into the system.

Acknowledgments

The authors thank the anonymous reviewers and our shepherd, Adrian Friday, for their many helpful suggestions and corrections. This work was supported in part through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

7. REFERENCES

[1] The Dartmouth wireless traces. `http://crawdad.cs.dartmouth.edu/data/dartmouth/`, 2005.

[2] MSN maps & directions. `http://maps.msn.com/`, 2005.

[3] The network simulator - ns-2. `http://www.isi.edu/nsnam/ns/`, 2005.

[4] Place lab - a privacy-observant location system. `http://www.placelab.org/`, 2005.

[5] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, July-September 2003.

[6] P. Billingsley. *Probability and Measure*, chapter 8, page 111. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons, third edition, 1995.

[7] J. Y. Le Boudec. Understand the simulation of mobility models with palm calculus. Technical Report EPFL/IC/2004/53, EPFL-DI-ICA, June 2004.

[8] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of Mobile Computing and Networking (MobiCom)*, pages 85–97, 1998.

[9] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2002.

[10] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 187–201. ACM Press, September 2004.

[11] K. Herrmann. Modeling the sociological aspects of mobility in ad hoc networks. In *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems (MSWiM)*, pages 128–129, New York, NY, USA, 2003. ACM Press.

[12] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Stochastic Processes*, chapter 1, page 1. Waveland Press, 1972.

[13] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of ACM/IEEE MSWiM'99*, pages 53–60, Seattle, WA, August 1999.

[14] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 217–229, San Diego, CA, September 2003. ACM Press.

[15] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[16] E. L. Lawler. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. In *Management Science*, volume 18, pages 401–405. March 1972.

[17] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*, chapter 3, pages 100–104. Dover Publications, February 2001.

[18] G. Lin, G. Noubir, and R. Rajaraman. Mobility models for ad hoc network simulation. In *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[19] Y. Lu, H. Lin, Y. Gu, and A. Helmy. Towards mobility-rich performance analysis of routing protocols in ad hoc networks: Using contraction, expansion and hybrid models. In *Proceedings of IEEE International Conference on Communications (ICC)*, June 2004.

[20] M. Musolesi, S. Hailes, and C. Mascolo. An ad hoc mobility model founded on social network theory. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 20–24, Venice, Italy, October 2004. ACM Press.

[21] W. Navidi and T. Camp. Stationary distributions for random waypoint models. In *IEEE Transactions on Mobile Computing*, volume 3, pages 99–108. 2004.

[22] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001.

[23] M. Sanchez. Mobility models. `http://www.disca.upv.es/misan/mobmodel.htm`, 2005.

[24] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive Wi-Fi mobility data. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1414–1424, March 2004.

[25] S. Tan, W. Lau, and A. Loh. Networked game mobility model for first-person-shooter games. In *Proceedings of the 4th Workshop on Network and System Support for Games (NetGames)*, Hawthorne, NY, October 2005. ACM Press.

[26] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 205–216, San Diego, CA, September 2003. ACM Press.