SMOOTH: A Simple Way To Model Human Mobility

Aarti Munjal, Tracy Camp, and William C. Navidi
Dept. of Math. and Computer Sciences
Colorado School of Mines
Golden, Colorado 80401
amunjal, tcamp, wnavidi@mines.edu

***Abstract*—In addition to being *realistic*, a mobility model should be easy to understand and use. Unfortunately, most of the *simple* mobility models proposed thus far are not *realistic* and most of the *realistic* mobility models proposed thus far are *not simple* to use. The main contribution of this work is to present SMOOTH, a new mobility model that is *realistic* and *simple* to use. SMOOTH is based on several known features of human movement.**

SMOOTH can also easily be tuned to imitate a recently published mobility model that is based on real GPS traces (SLAW). While SLAW was constructed by extracting statistical features of real human movement, it has input parameters that are very complex to set. SMOOTH, on the other hand, is *realistic* (e.g., SMOOTH is based on several known features of human movement) and is *simple* to use (e.g., SMOOTH does not have any complex input parameters).

We first present SMOOTH. We then validate that SMOOTH imitates SLAW by comparing several metrics obtained using the two mobility models. In addition, we compare the statistical features extracted from the synthetic traces generated by the two mobility models. The main contribution of our work is to provide the research community with a *simple*, *realistic* mobility model.

***Index Terms*—mobility model; mobile ad hoc networks; mobility modeling; simulation credibility; performance evaluation; realistic mobility; simulation scenarios.**

I. Introduction

Several mobility models have been developed in the past [17], [19]. Broadly, these mobility models can be categorized into *synthetic* and *trace-based* mobility models. The synthetic traces generated by a synthetic mobility model are not necessarily *realistic*. A *trace-based* mobility model, on the other hand, may not be *simple* to understand and use. To be widely adopted by the research community, a mobility model should be easy to comprehend and use. For example, many researchers know the RWP (Random Waypoint) mobility model is unrealistic [3], but it continues to be used due to its simplicity. In addition, a simple structure makes it easy to analyze the model mathematically [4]. The contribution of our work is in developing a *simple* mobility model that is *realistic*; our model is realistic since it generates synthetic traces that match well with real human movement. Other *simple* mobility models proposed thus far do not mimic real human movement [19] and other *trace-based* mobility models proposed thus far are not *simple* to use.

To validate that our mobility model, SMOOTH, imitates real human movement, we compare the synthetic traces generated by SMOOTH with the synthetic traces generated by SLAW. The statistical features extracted from these synthetic traces match well with the features extracted from the real GPS traces used in [2]. SLAW was constructed based on the real GPS traces collected from five different outdoor sites. After analyzing these GPS traces, SLAW was developed to capture the *self-similarity*[1] of visit points (a.k.a. waypoints) in the network. One of the input parameters to SLAW is the *hurst* parameter, which controls the degree of *self-similarity*. Our previous work in [18] and Lee et al. in [2] show that the *hurst* value has a significant effect on the performance metrics of a routing protocol. Although the *hurst* parameter is well-defined mathematically, it is highly difficult to estimate from a given data sample. There are a variety of estimation techniques available for the *hurst* parameter [20]; however, one is likely to obtain different *hurst* value estimates from different estimators on the same data set. In addition, any trends, noise or other forms of corruption in data (e.g., biased data) might lead to misleading results [20]. SMOOTH, on the other hand, can be easily tuned to imitate the *hurst* effect used to model the *self-similarity* in SLAW. Our algorithm is a very simple scheme that uses the number of clusters in the network as an input.

We offer researchers SMOOTH in order to leverage the statistical features present in real human movement in a simple and easy to understand manner. The main contribution of this work is to provide a *simple* mobility model, SMOOTH, that is *realistic* (i.e., SMOOTH generates synthetic traces that match the statistical features of real human movement).

The rest of the paper is organized as follows. In Section II, we briefly describe the recent work that has been done in this area. Section III describes SMOOTH, our new mobility model in detail. In Section IV, we provide a precise description of SLAW, a model based on real GPS traces. Section V contains details on how to tune SMOOTH to get SLAW. We also validate the realism of our mobility model using the synthetic traces generated by the SLAW mobility model. In Section VI, we provide the results of a performance evaluation of various

[1]A process is called *self−similar* if the aggregated processes (i.e., the processes obtained by averaging the original process over non-overlapping blocks) are highly correlated. Specifically, the autocorrelation function of a self-similar process is non-summable (i.e., ∞). For further details, see [5].

DTN routing protocols when SMOOTH is used. Lastly, we present our conclusions in Section VII.

II. Related Work

While synthetic mobility models are simple to use, they may not generate synthetic traces that mimic real human movement. Thus, the research community has recently focused on developing trace-based models (i.e., models that are constructed from real mobility traces). In trace-based models, real human movement traces are collected from various scenarios (e.g., campus, conferences, offices). The collected traces are then analyzed to identify the key characteristics of real human walks. Using this information, a model is constructed and is called a *trace-based* mobility model. The model is expected to generate synthetic traces that match well with the real human walk traces analyzed.

Different trace-based models developed are found in [9]–[12], [14]. In [15], the authors developed a simple mobility model, SWIM, that generates synthetic traces representing the social behavior among mobile nodes. The authors validated the synthetic traces generated by SWIM using the real GPS traces collected from three different experiments. In SWIM, every possible destination is assigned a calculated weight that increases with its popularity and decreases with its distance from the current GPS location of the mobile node. Two input parameters, K and α, are used for this purpose. The synthetic traces generated by SWIM seem to match the real GPS traces well. The GPS traces used for validation, however, are only from a campus and a conference scenario; thus, these traces do not capture the movement patterns from other scenarios, e.g., cities. In addition, the parameters K and α need to be set differently in order to match the real GPS traces. Also, the authors do not mention the flight distribution of mobile nodes, and this distribution appears to be a significant feature present in real human walk [2], [9].

In [2], the authors developed a mobility model, SLAW, that is based upon the real GPS traces collected from five different outdoor sites. The GPS traces used in SLAW are from diverse scenarios (two campuses, one state fair, one amusement park, and one metro city scenario); however, the synthetic traces generated by the SLAW mobility model heavily depend upon an input parameter called *hurst*. Our previous work in [18] and Lee et al. in [2] show that the *hurst* value has a significant effect on the performance metrics of a routing protocol. For example, in [18], we developed models that help researchers to generate rigorous simulation scenarios and these models were developed with the SLAW mobility model. During our analysis, we discovered SLAW's input parameters that most significantly affect the performance of a routing protocol. Specifically, we showed that the number of *nodes*, simulation *area* size, and the *hurst* value are the three input parameters that highly affect the performance of a routing protocol when used with SLAW. Unfortunately, as discussed in Section I, it is difficult to estimate the *hurst* value for a given data set. To illustrate, consider the following. The number of participants used in each outdoor site in [2] is small compared to the size of the measurement sites. Specifically, the measurement sites are of the order of several thousand meters squared and the number of participants used in each site ≤ 40. With such a difference in area size versus the number of participants, the *hurst* values extracted from the respective measurement sites may be biased. For example, we note that the site with the highest number of participants (i.e., Kaist with 34 participants) has the highest *hurst* value. In addition, due to the inherent complexity in techniques used for estimating *hurst* from a given data set and the possible biased trends that might be present in the measurements, one might find it complex to use SLAW the way one intends.

III. SMOOTH: Our Simple and Realistic Mobility Model

In the evaluation of real human walks, researchers have found several statistical features that exist [6]–[8], [13], [14], [25]. Specifically,

- **feature1:** The flights (i.e., a straight-line distance covered between two consecutive waypoints) distribution of mobile nodes follows a truncated power-law (TPL)[2].
- **feature2:** The ICTs (inter-contact times) (i.e., the amount of time between two successive contacts of the same pair of nodes) distribution of mobile nodes follows TPL.
- **feature3:** The pause-times (i.e., the amount of time a node pauses at a waypoint) distribution of mobile nodes follows TPL.
- **feature4:** Mobile nodes visit popular waypoints in the network.
- **feature5:** While moving, mobile nodes usually (but not always) visit the closest waypoint first.
- **feature6:** The distribution of mobile nodes is non-uniform in the network.
- **feature7:** Mobile nodes with common interests form communities and only tend to move around their communities of interests.

We capture all seven of these features in SMOOTH. Specifically, SMOOTH first creates communities with different levels of popularity, i.e., a few communities are more popular than the other communities. A mobile node in SMOOTH visits communities based upon their popularity level and, thus, meets a few mobile nodes more often than others. A mobile node in SMOOTH does not move around the simulation area randomly; instead, while moving, a mobile node is more likely to visit locations that are nearby to its current GPS location. We next describe how we created these communities in SMOOTH. We then show how SMOOTH meets **feature1** - **feature7** of real human walks.

A. Creating Communities

When evaluating real GPS traces (e.g., traces in [2]), it is evident that mobile nodes do not move randomly over the simulation area. Specifically, a mobile node visits a few

[2]The truncated power-law distribution follows power-law upto a certain time after which it is truncated by an exponential cut-off.

(i.e., popular) waypoints more frequently than waypoints that are less popular. Such movement patterns of mobile nodes represent the social behavior of humans and, thus, form communities. In SMOOTH, we represent communities by clusters, where popular communities are denoted by clusters of bigger sizes. (A *cluster-size* is the number of waypoints that belong to the cluster.) In SMOOTH, clusters are formed per the following technique.

We assume that the total number of waypoints, *N*, and the total number of clusters, *C*, available for a mobile node to visit are fixed and are specified as user inputs. *N* is divided into *C* clusters of unequal sizes. The cluster sizes are unequal to mimic realistic scenarios (i.e., popular communities). For example, students spend more time in an academic building (including classrooms) than they spend in a food court. Similarly, within a building, there are places that are visited more often than others (e.g., classrooms are visited more often than a conference room). Therefore, within a cluster, we plot groups (of random sizes). Specifically, to plot the waypoints of clusters and groups (within a cluster), we use the following approach:

1) for every cluster:
    a) for first group in this cluster:
        i) Plot the first waypoint uniformly over the simulation area with the following caveat; i.e., no two waypoints that belong to different clusters are within each other's transmission range.
        ii) Plot other waypoints that belong to this group within 0.1R (transmission range) of any of the waypoints that belong to this group.
    b) for every other group in this cluster:
        i) Plot the first waypoint such that it is within distance *d* from the last waypoint plotted for the previous group, $\frac{Y}{4} \leq \text{d} \leq \frac{Y}{3}$, where

$$Y = \frac{2Z}{C}, \tag{1}$$

Z is the side of the simulation area and *C* is the total number of clusters in the network. The area spanned by a cluster depends upon the simulation *area* size and the total number of clusters in the network. For example, if a user wants to create a single community, then SMOOTH plots it as one large cluster with various groups of random sizes. Thus, our choice of *d* (to separate random groups) depends upon the simulation *area* size (i.e., *Z*) and the total number of clusters (i.e., *C*).

B. SMOOTH: A Simple Algorithm

In this section and the next, we describe how SMOOTH satisfies all the seven statistical features (**feature1** - **feature7**) found in real human walks. Algorithm 1 describes an easy step-by-step working of SMOOTH. Further details on the algorithm follow.

In SMOOTH, a mobile node chooses a cluster (i.e., a community) with probability $\propto$ to its size (Step 3), i.e., a bigger cluster is chosen more often than a smaller cluster and, thus, more mobile nodes visit the bigger clusters making them more popular (**feature4** and **feature6**). From the selected cluster, a mobile node (uniformly) chooses a subset of waypoints to visit (Step 4). Since a bigger cluster has more waypoints, a mobile node chooses more waypoints to visit in a bigger cluster than a smaller cluster and, thus, spends more time in bigger clusters. Since every mobile node chooses a cluster with probability $\propto$ to its size, a few mobile nodes meet each other more often than others (**feature2** and **feature7**). While moving, a mobile node tends to visit the closest waypoint (i.e., the waypoint closest to its current GPS location); however, it may occasionally visit a farther location first (e.g., to attend a class that starts in a few minutes). To simulate this movement, the authors of [2] use a distance parameter *alpha* that determines the probability of choosing the next waypoint via the LATP (Least-Action Trip Planning) algorithm. The LATP algorithm uses a distance function (i.e., $\frac{1}{d^{\alpha}}$) to calculate the probability of selecting the next waypoint to visit; *d* is the distance between the current waypoint and the unvisited waypoint and α is the value of parameter *alpha* that controls the selection of the next waypoint. For α = 0, the next waypoint is selected randomly; for $\alpha = \infty$, the next waypoint to visit is the closest waypoint. In SMOOTH, we follow the same approach; that is, we use the LATP algorithm to select the next waypoint to be visited by the mobile nodes (**feature1** and **feature5**). For details, please refer to Section III.C of [2]. Pause-times are distributed according to truncated power-law (Step 6), i.e., mobile nodes pause for a long time at a few waypoints and pause for a short time at the majority of waypoints visited (**feature3**). The input parameters to SMOOTH follow.

1) *nodes* is the total number of mobile nodes in the network.
2) *area* is the size of the simulation area.
3) *waypoints (N)* is the total number of waypoints in the network that mobile nodes can visit.
4) *range (R)* is the transmission range of a mobile node[3].
5) *clusters (C)* is the total number of clusters (to be plotted) in the network.
6) *percent_waypoint (y)* is the percentage of waypoints visited by a mobile node from the selected cluster.
7) *alpha* controls the selection of the next waypoint visited

Algorithm 1 SMOOTH: pseudocode

1. Divide the simulation area into several communities (see Section III-A for details).
2. **for** each mobile node **do**
3. Select a community with probability $\propto$ to its *size*.
4. Select a subset (y%) of waypoints to visit from the selected community.
5. Visit the selected waypoints via the LATP algorithm (see Section III-B for details).
6. At each waypoint, pause for a *pause-time* distributed by power-law.
7. **end for**

[3]Many mobility models do not require the transmission range. SMOOTH needs the transmission range to form communities.

by a mobile node in the LATP algorithm.

8) *beta* is the levy exponent used for the pause-time distribution (i.e., describes the asymptotic behavior of the distribution).
9) (*min_pause, max_pause*) is the minimum and the maximum allowed pause-time (in seconds) for a mobile node.

C. Power-law flights and ICTs

In this section, we validate that our simple technique (described in Section III-A) for creating communities generates synthetic traces that preserve the statistical features observed in real human movement. Specifically, we validate that SMOOTH generates synthetic traces that have flights and ICTs (inter-contact times) distributed according to truncated power-law (i.e., **feature1** and **feature2**). The distribution of flights is of interest, as it helps to predict the trajectories of human walks. Similarly, the ICTs distribution is important, as it represents how often the mobile nodes meet each other and, thus, predicts the probability of forwarding a packet during routing. Recent studies [8], [13] have shown that flights (of mobile nodes) and ICTs (among a pair of mobile nodes) follow truncated power-law.

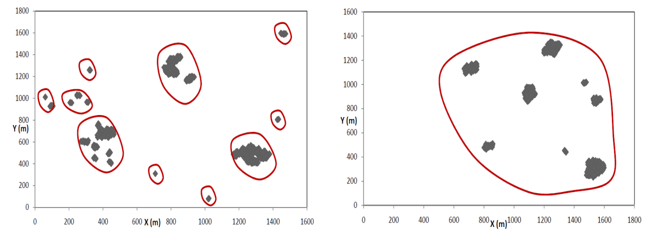
To validate that SMOOTH preserves the statistical features of real human movement, we simulated the two scenarios listed in Table I. (We choose only two scenarios due to space constraints; however, we note that the results are the same regardless of the scenarios chosen.) Figure 1 shows the synthetic maps of waypoints generated by SMOOTH for these two scenarios; the simulation area size and the total number of waypoints are kept constant to illustrate the clustering effect. Specifically, as shown, Scenario I represents 10 communities and, thus, is shown as various clusters distributed over the simulation area. Scenario II, on the other hand, represents a single community and is, thus, represented by a few groups over the simulation area. We extracted the distributions of flights and ICTs among nodes for both of these scenarios. Figure 2 shows the CCDFs (Complementary cumulative distribution functions) of flights and ICTs generated by SMOOTH for the Scenario I listed in Table I (satisfies **feature1** and **feature2**). Results for Scenario II are similar.

Parameter	Scenario1	Scenario2
Nodes	20	10
Area (m^2)	1600x1600	1600x1600
Clusters	10	1
Waypoints	1000	1000

TABLE I
EXAMPLE SCENARIOS SIMULATED ON SMOOTH.

IV. SLAW: A MODEL BASED ON REAL GPS TRACES

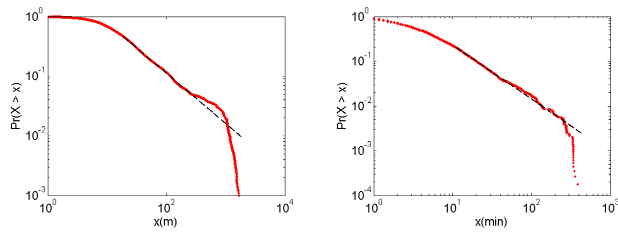
The SLAW mobility model was developed using the real GPS traces collected from five outdoor sites [2]. An analysis of SLAW concludes that SLAW meets **feature1** - **feature7**. SLAW also models the following statistical feature of real human movement:



Synthetic map of waypoints for Scenario I

Synthetic map of waypoints for Scenario II

Fig. 1. Synthetic map of waypoints generated by SMOOTH for the scenarios listed in Table I. The circled area illustrates the communities in each scenario.



CCDF of flights

CCDF of ICTs

Fig. 2. CCDFs of flights and ICTs extracted from the synthetic trace generated by SMOOTH for Scenario I listed in Table I.

- **feature8:** The distribution of waypoints over the simulation area is *self-similar*.

The synthetic mobility traces generated by SLAW are expected to be similar to the real GPS traces used in its development; however, using the SLAW mobility model can be extremely complex. For example, the waypoint placement technique used in SLAW to meet **feature8** requires a complete understanding of the *hurst* parameter. The real GPS traces used in [2] were analyzed to find the *hurst* values for each of the five outdoor sites. The *hurst* value is used to control the degree of self-similarity; a greater *hurst* value represents a higher degree of self-similarity in the network.

Algorithm 2 presents SLAW; *x*, *y*, and *N* are user inputs to the SLAW mobility model. The waypoints are distributed over the simulation area (Step 1) in a self-similar manner (**feature8**) using a technique related to Brownian Motion [16]. The *hurst* parameter can take values within a pre-defined range $0.5 < hurst < 1$; for a *hurst* value close to 1, the process is highly self-similar. In the context of waypoints' placement, a higher *hurst* value means that the network looks more self-similar at different scales than with a lower *hurst* value.

As shown in our previous work [18], a network remains mostly partitioned in SLAW. The high amount of network partitioning is due to two reasons: *inter-cluster partitioning* and *intra-cluster partitioning*. Specifically, in SLAW, clusters are formed (Step 2) via *transitive closure*, i.e., the waypoints

Algorithm 2 SLAW: pseudocode

1. Distribute N waypoints over the simulation area in a *self-similar* manner (controlled by the *hurst* parameter).
2. Divide the distributed waypoints into clusters via *transitive closure*.
3. **for** each node **do**
4. Select x clusters with probability proportional to *cluster-weight*.
5. Randomly select y% of waypoints from each of the x selected clusters.
6. Visit all selected waypoints (i.e., one trip) via the LATP algorithm (see Section III-B for details).
7. At each waypoint, pause for a *pause-time*.
8. Randomly replace one of the x clusters after every trip.
9. **end for**

that are connected to each other through multiple links form one cluster [2]. Due to this cluster-formation technique, mobile nodes visiting waypoints that belong to different clusters are partitioned and, thus, contribute to the *inter-cluster partitioning*. Similarly, within a cluster, a mobile node chooses a subset of the waypoints to visit; the waypoints selected by a mobile node to visit may not be connected. Therefore, within a cluster, the mobile node pairs that have no communication routes among them are unable to communicate and, thus, contribute to the *intra-cluster partitioning*.

The *cluster-weight* (Step 4) is defined by the number of waypoints in a cluster. Since clusters are of different weights, a cluster with a higher weight is selected more often than a cluster with a lower weight. Once a mobile node chooses a subset of waypoints to visit (Step 5), it visits them according to the LATP algorithm (Step 6). Pause-times are distributed according to truncated power-law (Step 7), i.e., mobile nodes pause for a long time at a few waypoints and pause for a short time at the majority of waypoints visited.

Mobile nodes in SLAW tend to move around predefined areas of mobility. For example, a node replaces (uniformly and randomly) one cluster after every trip (Step 8). Therefore, a node spends more time in a few clusters than others and, thus, has shorter ICTs with the mobile nodes that are moving within these clusters. The goal of SLAW is to capture statistical features of real human movements. Next, we describe how to imitate SLAW with SMOOTH, and with much less effort and complexity.

V. SMOOTH Can Imitate SLAW

Both SMOOTH and SLAW were constructed by extracting statistical features present in real human movement. SMOOTH was created via several known features of real human movement. SLAW, on the other hand, was created from analysis of real GPS traces collected from five different outdoor sites. The GPS traces used in the development of SLAW were collected from sites that represent diverse scenarios. To develop a mobility model that can cater the needs of a wide range of users, the mobility traces used in its development should be collected from different scenarios (e.g., city, campus, offices).

The synthetic traces generated by SLAW seem to match well with the real GPS traces used in its development. To validate that SMOOTH generates synthetic traces that can mimic real human movement in diverse scenarios, we compare the synthetic traces generated by these two mobility models. First, since the synthetic traces generated by SLAW heavily depend upon the associated *hurst* value, we discuss how to tune SMOOTH to imitate the *hurst* effect in this section. We then validate that SMOOTH can mimic SLAW.

A. The Hurst Effect

As mentioned in Section I, it is difficult to estimate the *hurst* value from a given data set. Specifically, there are various techniques that have been developed to estimate the *hurst* value from a given data sample. These techniques, however, require a full understanding of both the estimation technique and the characteristics of the data set involved in the estimation process [20]. For example, the real GPS traces used in the development of SLAW have been collected from five different outdoor scenarios. Table II lists these five scenarios and their respective *hurst* values. These values have been deduced by using the *aggregated* variance technique of estimating the *hurst* parameter for a given data set [9]. Specifically, for a given scenario (e.g., Kaist in Table II), the waypoints registered in every trace are plotted. The simulation area is then divided into squares of the same size (i.e., 5mx5m); we refer to the side of the square as u. The normalized count of waypoints in each square (i.e., the number of waypoints/square area) and the variance among these normalized counts is calculated. The process is then repeated using a bigger value of u (e.g., 25m). The variance among these normalized counts for each value of u is plotted on a log-log scale. If the plotted variance decays no faster than -1, then the data sample has the value of *hurst* = 1 - β, where β is the absolute value of the slope of the plotted variance.

Scenario	Waypoints	Area	Hurst	Hours	Clusters (C)
Kaist	10598	10256x18650	0.80	1445.15	459
NCSU	3316	2586x2347	0.65	310.62	24
NYC	1105	31432x18900	0.55	293.21	127
State Fair	691	1141x995	0.75	47.01	3
Disney	4085	8214x9446	0.75	378.08	236

TABLE II
Original scenarios extracted from the GPS traces used in [2].

In SLAW, the precision of the waypoints' placement technique increases with an increase in the number of levels. We now show that tuning SMOOTH's random groups placement technique accurately imitates the *hurst* effect, and, thus, satisfies **feature8**. We explain our modified random groups placement technique next.

B. A Simple Technique To Imitate the Hurst Effect

As discussed in Section II, our previous work in [18] shows that the *hurst* value in SLAW significantly affects the following three metrics of interest:

- A_{sp}Hops: Average shortest-path hop count

- ANP: Average network partitioning
- ANC: Average neighbor-count

Specifically, for a given pair of values of *area* and number of *nodes*, ANP and ANC are maximum for the lowest *hurst* value in SLAW (close to 0.5). Similarly, A_{sp}Hops is maximum for the highest possible *hurst* value in SLAW (close to 1.0). To imitate the SLAW *hurst* effect accurately in SMOOTH, we can control the following two characteristics of the random groups plotted per cluster:

- *The maximum size (s)* of a group controls the ANC in the network and
- *The maximum distance (x)* of a waypoint from other waypoints in the group controls the metrics (A_{sp}Hops, ANP) in the network.

Within a given SMOOTH cluster, the first group is plotted by choosing a reference point (uniformly and randomly distributed over the simulation area); this reference point is the first plotted waypoint. In our modified SMOOTH (to mimic SLAW), we add other waypoints to the group within x meters of any of the waypoints that belong to this group. We also restrict the maximum size of a random group to s. As discussed previously, the other groups of waypoints in this cluster are plotted by choosing a new reference point, such that its distance d from the last waypoint plotted for the previous group is $\frac{Y}{4} \leq d \leq \frac{Y}{3}$. As a reminder, Y is a constant that depends upon the size of the simulation area (i.e., Z) and the total number of clusters (i.e., C) in the network.

To imitate SLAW, we choose (x,s) according to the *hurst* value. See Table III for details. Controlling these two characteristics of a random group in a cluster accurately imitates the degree of self-similarity controlled by the *hurst* parameter. *'NA'* in Table III denotes that there is no limit on the maximum size of a random group for this *hurst* value (e.g., 0.55); as discussed, ANC is maximum for *hurst* = 0.5. We note that we used a constant transmission range (R=100m) in our work, which is consistent with the work in [2].

Hurst	Size (s)	Distance (x in meters)
$0.55 \leq hurst < 0.6$	NA	5
$0.60 \leq hurst < 0.65$	50	10
$0.65 \leq hurst < 0.70$	30	20
$0.70 \leq hurst < 0.85$	30	50
$0.85 \leq hurst \leq 0.95$	20	80

TABLE III
MAXIMUM SIZE AND DISTANCE DEFINED FOR MOST *hurst* VALUES.

1) Number of Clusters: One of the inputs to SMOOTH is the total number of clusters in the network, denoted by C. The number of clusters in SLAW depends upon the simulation *area* size and the *hurst* parameter. In this work, we calculated this parameter by executing several iterations of SLAW and averaging the number of clusters generated for a given scenario. In other words, for a given pair of values for simulation *area* and *hurst* values, we ran several iterations of SLAW and calculated the number of clusters generated in each iteration. We then averaged these results for all the iterations and calculated the parameter C (with a confidence interval of 95%).

We provide an easy to use equation to calculate C for a given choice of *area* and *hurst* values. In Table II, we note that the number of clusters for the NYC scenario is far less than the number of clusters for the Kaist scenario, even though NYC has a larger simulation *area* size and a much lower *hurst* value. We also noted that the number of waypoints in the Kaist scenario is ten times the number of waypoints in the NYC scenario. Thus, in our regression analysis (described next), we use the number of waypoints as one of the input parameters.

We simulated SLAW for all 125 possible combinations of the values listed in Table IV and calculated the parameter C. Specifically, we ran 200 iterations of SLAW for each of the 125 possible combinations. We note that the combinations cover a large space of possible SLAW inputs. The number of clusters obtained in all the iterations were then averaged. (All the results used in this analysis are with a confidence interval of 95%.) Using these averaged results for the number of clusters, we then fit a linear model for the parameter C via linear regression. The fitted linear model for the parameter C is given by equation 2. The coefficient of determination for this model (i.e., equation 2) is 94% and, thus, provides a very good estimate of the number of clusters that SLAW will generate for a given pair of values of *area* and *hurst*.

Parameter	Values
Area	1000, 1200, 1500, 1800, 2000
Hurst	0.55, 0.65,... 0.95
Waypoints	800, 1100, 1400, 1700, 2000

TABLE IV
SLAW PARAMETERS WITH A REQUIRED RANGE OF VALUES

$$\begin{aligned} \mathrm{C} = \quad & 37.0 + 0.205\ area \\ & - 0.228\ area * hurst \\ & + 77.4\ hurst\_sq - 110\ hurst \end{aligned} \tag{2}$$

C. Validation Results

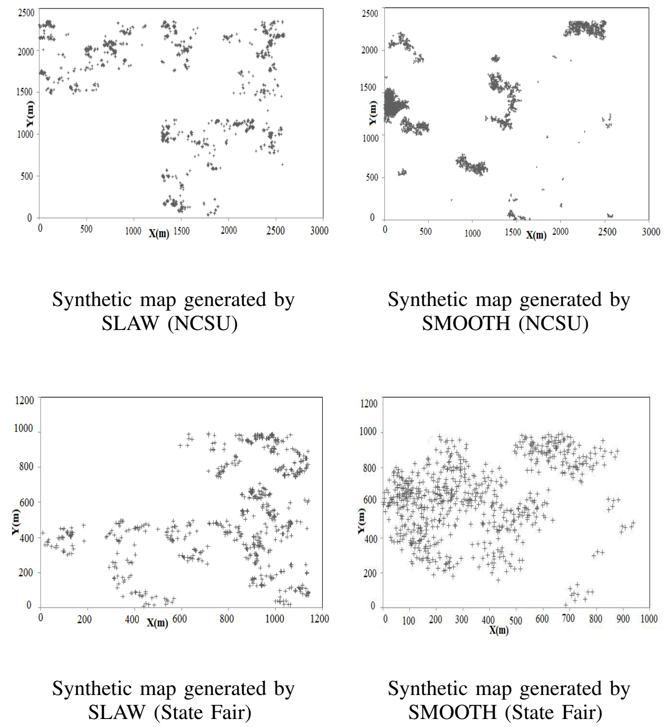
SLAW is based upon real GPS traces collected from diverse scenarios. The synthetic traces generated by SLAW seem to match well with these real GPS traces. Thus, to validate that SMOOTH generates synthetic traces that can mimic SLAW and real human movement, we compare the synthetic traces generated by SMOOTH with the synthetic traces generated by SLAW. Specifically, we compare SMOOTH and SLAW for the following criteria.

- The values obtained for three metrics (A_{sp}Hops, ANP, ANC). To obtain, we simulate the five original scenarios used in the development of SLAW [2].
- Statistical features extracted from the synthetic traces generated for the five original scenarios.

1) Metric Values: As discussed previously, in SLAW the placement of waypoints over the simulation area is controlled by the *hurst* parameter. The original GPS traces used in the development of SLAW were analyzed to estimate the *hurst* values for their respective sites. In this section, we use the five original scenarios used in [2], [9] and compare the synthetic traces generated by SLAW and SMOOTH for the metric values and the statistical features extracted (i.e., flights and ICTs distribution). Table II lists these five scenarios. Figure 3 shows the synthetic maps of waypoints generated by SLAW and SMOOTH for two of these five scenarios: the NCSU and the State Fair scenarios (the 2nd and the 4th scenario in Table II), respectively. (The synthetic maps generated by SMOOTH for the other three scenarios are similar; they are not included due to space constraints.) The synthetic maps generated by SLAW are plotted using the *hurst* values extracted from their respective GPS traces used in [9]. The synthetic maps generated by SMOOTH are plotted using our simple scheme described in Section V-B. Specifically, for a given *hurst* value, we calculated the values of (C,x,s) and simulated SMOOTH for these values. We note that even though SMOOTH does not use the precise waypoint placement technique as SLAW does (using the *hurst* parameter), SMOOTH accurately preserves the statistical features of real human movement. To illustrate, we compare the metric values and the CCDFs of flights and ICTs generated by SLAW and SMOOTH, and validate that our simple waypoint placement technique used in SMOOTH imitates the *hurst* effect in SLAW.

We compared SLAW and SMOOTH for the values obtained for the following three metrics: A_{sp}Hops, ANP, and ANC. To obtain these values, we simulated the scenarios listed in Table II. For every scenario, we simulated 200 independent simulation runs. We discarded the first 50 hours of simulation time to avoid any initialization bias in the simulation results; discarding 50 hours is consistent with the work in [2]. During every simulation run, we took a snapshot of the network at the 51^{st} hour of the simulation time. We calculated the metric values during every snapshot. We then averaged the metric values over all the snapshots taken. We simulated SLAW for these scenarios using the *hurst* values extracted from their respective GPS traces. In SMOOTH, we first calculated the number of clusters for each of these five scenarios. (See Section V-B1 for details.) We then simulated SMOOTH for these five scenarios using (*C, x, s*) as input parameters. (See Sections III and V-B for details.) Table V lists the values obtained for these three metrics from both mobility models; all values listed were calculated using a confidence interval of 95%. As shown in Table V, SLAW and SMOOTH produced similar metric values for the five scenarios. For a list of values used for the other simulation parameters (kept constant) during simulation, see Table VI. For details about these input parameters, see Section III.

2) Statistical Features: SLAW is constructed by extracting statistical features of real human movement from real GPS traces. In [2], SLAW was validated by comparing the statistical features extracted from its synthetic traces with the features



Synthetic map generated by SLAW (NCSU)

Synthetic map generated by SMOOTH (NCSU)

Synthetic map generated by SLAW (State Fair)

Synthetic map generated by SMOOTH (State Fair)

Fig. 3. Synthetic maps of waypoints generated for the NCSU and the State Fair scenarios listed in Table II.

		SLAW			SMOOTH		
Scenario	**Nodes**	**Hops**	**ANP**	**ANC**	**Hops**	**ANP**	**ANC**
Kaist	300	1.62	0.9917	0.6485	1.36	0.9923	0.5231
NCSU	200	2.02	0.8863	5.726	2.23	0.8587	6.5905
NYC	300	1.185	0.8419	20.28	1.025	0.8637	17.854
State Fair	150	3.867	0.6064	5.7485	3.634	0.5647	5.7172
Disney	200	1.6768	0.983	1.0054	1.385	0.9552	1.2465

TABLE V
COMPARISON OF METRIC VALUES OBTAINED BY SIMULATING THE SCENARIOS LISTED IN TABLE II ON SLAW AND SMOOTH.

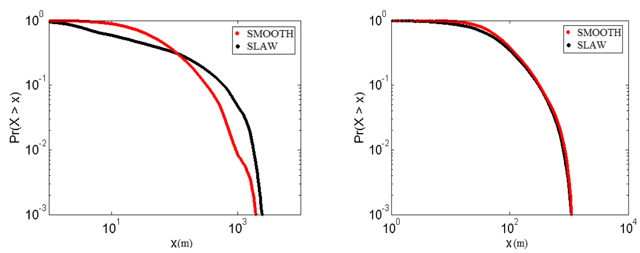
extracted from the real GPS traces. To validate that SMOOTH imitates real human movement, we compare the statistical features extracted from the synthetic traces generated by both of these mobility models. Specifically, we compare the two mobility models for the CCDFs of flights and ICTs for several synthetic traces from each model. The traces were generated for all the scenarios listed in Table II; however, due to space constraints, we only plot the CCDFs extracted for two of these scenarios (i.e., the NCSU and the State Fair scenarios listed in Table II). Figures 4 represents the CCDFs of flights obtained from the synthetic traces generated by SLAW and SMOOTH for the State Fair and the NCSU scenarios. As shown, the CCDFs of flights extracted from SMOOTH's synthetic traces match well with the CCDFs of flights extracted from SLAW's synthetic traces.

Similarly, we extracted the distribution of ICTs (among every pair of mobile nodes) for the scenarios listed in Table II.

Parameter	NCSU	State Fair
waypoints	3316	691
range (R)	$100m$	$100m$
percent_waypoints	0.10	0.10
alpha	1.5	3
beta	1.0	1.0
min_pause	$30(sec)$	$30(sec)$
max_pause	$42000(sec)$	$42000(sec)$

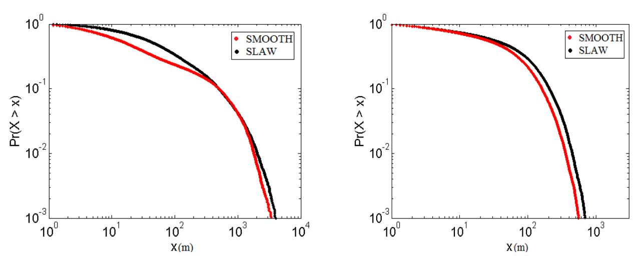
TABLE VI
INPUT PARAMETERS WITH THE VALUES USED FOR SIMULATION.

To compare the synthetic traces for the two mobility models, we plot the CCDFs of the ICTs for the NCSU and the State Fair scenarios (the 2nd and the 4th scenarios listed in Table II). Figures 5 represent the CCDFs of ICTs obtained from the synthetic traces generated by SLAW and SMOOTH for the State Fair and the NCSU scenarios. As shown, the CCDFs of ICTs extracted for the two mobility models match well.



CCDFs of flights (NCSU) CCDFs of flights (State Fair)

Fig. 4. CCDFs of flights extracted from the synthetic traces generated by SMOOTH and SLAW for the NCSU and State Fair scenarios listed in Table II.



CCDFs of ICTs (NCSU) CCDFs of ICTs (State Fair)

Fig. 5. CCDFs of ICTs extracted from the synthetic traces generated by SMOOTH and SLAW for the NCSU and State Fair scenarios listed in Table II.

VI. PERFORMANCE EVALUATION OF DTN ROUTING PROTOCOLS

The movement patterns of mobile nodes significantly affect the performance of a routing protocol. Thus, we investigate how a routing protocol performs on SMOOTH in this section. For this purpose, we simulated three different routing protocols on SMOOTH. In [2], two different classes of DTN routing protocols (i.e., *memoryless* and *memory-full*) were simulated on the synthetic mobility traces generated by SLAW. In a memoryless routing protocol, a mobile node waits until it meets the destination node (of the packet) and, thus, the packet is routed in a single hop. In a memory-full routing protocol, on the other hand, a mobile node uses its past contact information (with other mobile nodes) in making its routing decisions. Specifically, a mobile node keeps timestamps that denote its past meeting times with the other mobile nodes. In [2], the authors conclude that since SLAW can predict the distribution of ICTs among a pair of mobile nodes well, a memory-full routing protocol performs better on SLAW than on other random mobility models. In other words, the regularity of trips in the SLAW mobility model helps mobile nodes make better decisions regarding packet forwarding.

To validate SLAW [2], five DTN routing protocols (i.e., Random Forwarding [21], Direct Transmission [21], PRoPHET [22], LET [23], and ECT [2]) were evaluated over the synthetic mobility traces generated by SLAW. LET [23] refers to a metric called *Last Encounter Time*, in which a mobile node chooses the neighbor node that has most recently met the destination to relay a packet for that destination. ECT was introduced by the authors of SLAW and refers to a metric called *Expected Contact Time*. ECT is computed as the difference between LET and ICT for a pair of mobile nodes and denotes the expected time remaining before their next encounter. Thus, a neighbor node with a lower value of ECT is chosen as a relay for the packet, as it is more likely to meet the destination sooner than other mobile nodes. To validate that SMOOTH preserves the regularity in human walks and predicts the ICTs of a pair of mobile nodes similar to SLAW, we simulated one memoryless and two memory-full routing protocols used in [2]. Specifically, we compared the performance of Direct Transmission, LET, and ECT on the synthetic mobility traces generated by SMOOTH.

We simulated these three routing protocols on the ONE simulator for DTN protocol evaluation [24]. For this purpose, we generated synthetic mobility traces via SMOOTH for 50 mobile nodes. The movement traces were collected for 200 hours. To avoid any initialization bias, we discarded the first 50 hours of movement data. (Discarding 50 hours is consistent with the work in [2].) Source and destination pairs for packets are randomly chosen. During simulation, the speed for all mobile nodes is set to 1m/s (again consistent with the value for the node speed used in [2]). We simulated all three of these routing protocols for four different transmission range values. For simulation purposes, we used the original scenarios listed in Table II. Specifically, the values for the input parameters such as the total number of waypoints available for a mobile node to visit, the size of the simulation area, etc. were kept the same as found in the scenarios extracted from the real GPS traces used in SLAW. In [2], to evaluate the effect of the nodes' mobility on the performance of a routing protocol, the nodes maintained full historical information. Specifically, the entries of past contacts were not deleted even when they became stale. This procedure was also done in SLAW.

Figure 6 shows the delay measurements obtained by simulating these three routing protocols for the NCSU scenario. As mentioned previously, the size of the simulation area and the other parameters (e.g., the total number of waypoints available for a mobile node to visit, total number of clusters in the network etc.) are set to the same values as used in Section V-C for the NCSU scenario. As shown in the figure, SMOOTH differentiates well among the performance of these three DTN routing protocols. As shown, the Direct Transmission performs poorly, as a mobile node waits until it meets the destination node of the packet. LET performs better than Direct, as it uses its past contact information to forward packets. ECT, as concluded in [2] for SLAW, performs the best due to the metric used in making routing decisions. In ECT, a mobile node forwards its packet to a neighbor node that has the least remaining expected contact time with the destination and, thus, incurs a lower delay in routing a packet than other protocols. In summary, SMOOTH provides delay results that are very similar to the results provided by SLAW in [2].

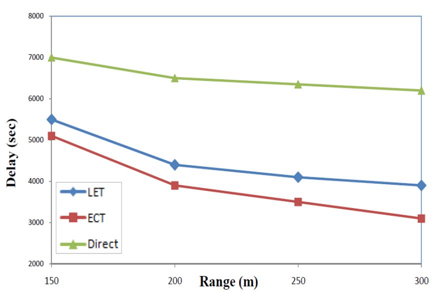


Fig. 6. Average routing delays for the three DTN routing protocols on SMOOTH.

VII. Conclusion

In this work, we developed a simple mobility model, SMOOTH, that was created using known features of human movement. To be widely adopted by the research community, a mobility model should be simple to understand and use. The waypoint placement technique used in SMOOTH is easy to understand and use. We validated our mobility model by comparing the statistical features extracted from its synthetic traces with the features extracted from the synthetic traces generated by SLAW. SMOOTH provides a very good match for these statistical features and is also able to predict the performance of a routing protocol, similar to SLAW.

Acknowledgment

We thank Stuart Kurkowski for allowing us to use his code for calculating the metrics. This work supported in part by NSF Grant CNS-0848130. Research group's URL is *http://toilers.mines.edu.* This site gives details on obtaining SMOOTH.

References

[1] S. Kurkowski, T. Camp, and W. Navidi. Scenario standards for rigorous MANET routing protocol evaluation. In *Proceedings of MASS 2006*.

[2] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. SLAW: A mobility model for human walks. In *Proceedings of INFOCOM 2009*.

[3] J. Yoon, M. Liu, B. Noble. Random waypoint considered harmful. In *Proceedings of INFOCOM 2003*.

[4] W. Navidi, T. Camp, and N. Bauer. Improving the accuracy of Random Waypoint simulations through steady-state initialization. In *Proceedings of the 15th International Conference on Modeling and Simulation (MS)*, pages 319-326, 2004.

[5] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic. In *Proceedings of the IEEE/ACM Transactions on Networking,* pages 1-15, 1994.

[6] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature,* pages 462-465, 2006.

[7] M. C. Gonzalez, C. A. Hidalgo, and A. Barabasi. Understanding individual human mobility patterns. *Nature,* pages 779-782, 2008.

[8] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy walk nature of human mobility. In *Proceedings of INFOCOM 2008*.

[9] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Demystifying levy walk patterns in human walks. In *Techical Report,* NCSU, 2008. http://netsrv.csc.ncsu.edu/export/Demystifying_Levy_Walk_Patterns.pdf Last visited 26 May, 2010

[10] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of MobiCom 2003*.

[11] J.Yoon, B. Noble, M. Kiu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *Proceedings of MobiSys 2006*.

[12] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling time-variant user mobility in wireless mobile networks. In *Proceedings of INFOCOM 2007*.

[13] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovic. Power law and exponential decay of inter contact times between mobile devices. In *Proceedings of MobiCom 2007*.

[14] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proceedings of INFOCOM 2006*.

[15] A. Mei and J. Stefa. SWIM: A simple model to generate small mobile worlds. In *Proceedings of INFOCOM 2009*.

[16] I. Norros, P. Mannersalo, and J. Wang. Simulation of fractional brownian motion with conditionalized random midpoint displacement. In *Advances in Performance Analysis,* pages 77-101, 1999.

[17] F. Bai and A. Helmy. A survey of mobility models. http://www.cise.ufl.edu/~helmy/papers/Survey-Mobility-Chapter-1.pdf Last visited 17 May 2010

[18] A. Munjal, T. Camp, and W.C. Navidi. Constructing rigorous MANET simulation scenarios with realistic mobility. In *Proceedings of the 11th European Wireless Conference,* 2010.

[19] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing: Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications,* pages 483-502, 2002.

[20] R. G. Clegg. A practical guide to measuring the *hurst* parameter. In *Proceedings of the 21st UK Performance Engineering Workshop, School of Computing Science Technical Report Series, CSTR-916,* University of Newcastle, 2006.

[21] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The single-copy case. *IEEE/ACM Transactions on Networking,* pages 63-76, 2008.

[22] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R),* pages 239-254, 2003.

[23] H. D.-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In *Proceedings of MobiHoc 2003*.

[24] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of SIMUTools 2009*.

[25] C. Song, Z. Qu, N. Blumm, and A. Barabasi. Limits of predictability in human mobility. *Science,* pages 1018-1021, 2010.