Steady-State of The SLAW Mobility Model

Aarti Munjal[1], William C. Navidi[2], and Tracy Camp[2]
[1] University of Colorado Denver, Aurora, CO, USA
[2] Colorado School of Mines, Golden, CO, USA
Email: aarti.munjal@ucdenver.edu; {wnavidi, tcamp}@mines.edu

Abstract —Movement patterns of mobile nodes significantly affect the performance of a Mobile Ad hoc Network (MANET) routing protocol. Thus, it is essential that the mobility traces generated by a mobility model closely match the trace data collected from experiments. In addition, the probability distribution of mobile nodes (over the simulation area) varies with time before it comes to a steady-state. In other words, distribution of mobile nodes in steady-state is independent of their initial position. If traces generated by a mobility model are used before its steady-state is reached, this variability in distribution may lead to misleading results, called *initialization bias*. Thus, for credible MANET simulations, it is required that a mobility model used (1) is realistic (i.e., its mobility traces closely match the experimental data) and (2) starts in a steady-state (i.e., there is no *initialization bias*). In this work, we analyze a recently published *realistic* mobility model, called SLAW. The SLAW mobility model is based on real GPS traces collected from five outdoor sites. Previous work with SLAW is done by discarding the first few hours of simulation time; however, discarding initial data does not guarantee that the model has reached its steady-state. Thus, the main contribution of our work is that we provide methods for sampling from the steady- state distributions of mobile nodes' locations and pause-times. Sampling from the steady-state distributions allows the SLAW mobility model to start in a steady-state and thus, avoids *initialization bias* related to the mobility model.

Index Terms—Stationary distribution, realistic mobility, mobility models, mobile ad hoc networks.

I. Introduction

MOBILITY models are broadly categorized into *synthetic* and *trace-based* mobility models [1]. While most of the synthetic mobility models proposed thus far are simple, the mobility traces generated by a synthetic mobility model do not closely match real trace data (e.g., GPS traces available at [2]). Thus, there is a growing interest in the development and use of mobility models that are based on real trace data collected from several indoor and outdoor sites. Mobility models based on real traces are called *trace-based* or *realistic* mobility models [3]. Since a *trace-based* mobility model model the statistical features present in human movement, synthetic traces generated by a *trace-based* mobility model closely match experiment data.

The movement patterns of mobile nodes significantly affect the performance of a MANET routing protocol [4]. Simulations are typically the only tools available for the performance evaluation of a routing protocol. While simulations offer several advantages over a real testbed/experiment (e.g., reproducibility of results, testing a network under diverse scenarios etc.), various assumptions made during simulations often lead to misleading results. For example, the movement patterns of mobile nodes might take sometime before it comes to a steady-state (i.e., the probability distribution of mobile nodes over the simulation area becomes independent of their initial position). Authors in [5] have showed that analyzing the synthetic traces generated by a mobility model before it reaches a steady-state results in an *initialization bias* (e.g., bias due to initial position of mobile nodes). Previous trends for avoiding bias in data collected from the synthetic traces include discarding the first few hours of simulation time, called *burn-time*. Measurements are only taken after the simulations have run for burn-time. However, as discussed in [5], this approach has several drawbacks. Specifically, discarding data that is generated for *burn-time* is not an efficient use of the simulation time. In addition, it is hard to predict how long the *burn-time* should be. Thus, using *burn-time* as a means of starting a mobility model in its "steady-state" is not recommended.

The main contribution of our work is that we develop a method for sampling from the stationary distributions for a recently published mobility model called SLAW. The SLAW mobility model is based on real GPS traces collected from five different outdoor sites. The synthetic traces generated by SLAW are expected to be close to real trace data, as it models various statistical features present in real human walk. Previous work with SLAW has discarded the first few hours of simulation time. However, discarding the first few hours of simulation time does not guarantee that the distribution has reached steady-state.

The rest of this paper is organized as follows. In Section II, we briefly describe the SLAW mobility model. Section III provides methods for sampling from the stationary distribution for SLAW with a zero pause-time (i.e., mobile nodes do not pause at any of the locations they visit). Implementation details and simulation results are then included for the scenarios simulated for SLAW without pause. In Section IV, we provide methods for

Manuscript received December 22, 2013; revised April 27, 2014.
This work was supported by the National Science Founda-tion under Grant No. CNS-0905513.
Corresponding author email: aarti.munjal@ucdenver.edu.
doi:10.12720/jcm.9.4.322-331

sampling from the stationary distribution for SLAW with a pause- time. Implementation details and simulation results are then included for the scenarios simulated for SLAW with pause. Finally, we state our conclusions in Section V.

II. THE SLAW MOBILITY MODEL

Various random mobility models proposed thus far can be found in [4], [6]. While these random mobility models are simple to use, their synthetic traces are unable to match various statistical features present in real human movement [7]–[9]. For an exhaustive evaluation of a rout- ing protocol, the movement patterns of mobile nodes in simulations should closely match the movement patterns of humans in the real world. For this purpose, there is a growing interest in mobility models that are either based on real trace data (collected from deployments at various sites) or are based on several known features of real human movement. As discussed in Section I, these mobility models are called *trace-based* mobility models. In this work, we consider a recently published mobility model, SLAW, based on real GPS traces. The GPS traces used in SLAW are collected from five different scenarios including two campuses, a metro city, Disney World, and a state fair scenario [7]. Next, we briefly describe the SLAW mobility model.

A. Movement of Mobile Nodes in SLAW

Algorithm 1 presents SLAW. Further details on the algorithm follow. In SLAW, the total number of waypoints (i.e., locations to visit by a mobile node) in SLAW is fixed. From the analysis of the real GPS traces used in SLAW, the authors concluded that the distribution of waypoints can be modeled as a *self-similar* process (**Step1**). 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 [10]. The degree of self-similarity of waypoints over the simulation area is controlled by a parameter called *hurst* [11]. The *hurst* parameter can take values within the interval [0.5, 1]; for a hurst value < 0.5, a process is not self-similar. After waypoints are placed, clusters are formed via *transitive* closure (**Step 2**). (See [7] for details).

At the start of simulation, every mobile node chooses a set of clusters (say x) and a fraction (say y%) of waypoints to visit from each of the selected clusters (Step 4 and Step 5). A mobile node visits all of the the selected waypoints per the LATP (Least Action Trip Planning) algorithm (Step 6). Under LATP, a mobile node chooses the next waypoint k, to visit with a probability proportional to , where

- d_k is the distance between the mobile node's current location and waypoint k and
- *Alpha* is a positive constant (see Section II-B for details).

Algorithm 1 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*, the number of waypoints in the cluster.
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 II-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**

At every waypoint, the mobile node pauses for a pausetime (**Step 7**) that is power-law distributed (see Section IV for details). Once the mobile node has visited all the selected waypoints from this set of clusters (i.e., one trip), it randomly replaces one of the x clusters (**Step 8**) and again chooses a subset of waypoints from the new set of x clusters (two of which are the same as in the previous trip). A list of input parameters to SLAW follows:

- *Alpha* is a parameter that controls the choice of the next waypoint to be visited (in LATP algorithm).
- *Nodes* is the total number of nodes in the network.
- *Area* is the size of the network.
- *Wps* is the total number of waypoints in the network.
- *Hurst* is the parameter that controls the self- similarity of waypoints in the network.
- *Range* is the transmission range of a node in the network.
- *Per_waypointthe* is the percentage of waypoints that a mobile node chooses to visit from each cluster.
- *Clusters* is the number of clusters that a mobile node visits in the network.
- *Beta* is the parameter used for the pause-time dis- tribution.
- p_{min} is the minimum allowed pause-time.
- p_{max} is the maximum allowed pause-time.

B. Effect of Alpha

As discussed in Section II-A, *alpha* is an input parameter to the LATP algorithm that controls the selection of the next waypoint to be visited by a mobile node. Based on the value of *alpha*, the algorithm calculates the probability of selection for every waypoint that needs to be visited by a mobile node. Specifically, the probability of selecting an unvisited waypoint is proportional to a weight assigned to the waypoint and is given by:

$$p_s = \frac{\left(\frac{1}{d_s}\right)^{alpha}}{\sum_{k=1}^{V}\left(\frac{1}{d_k}\right)^{alpha}} \quad (1)$$

- p_s is the probability to select an unvisited waypoint s,
- d_k is the distance between a mobile node's current waypoint and an unvisited waypoint *k*, and
- V is the total number of unvisited waypoints for the mobile node.

Thus, for *alpha* = 0, all the unvisited waypoints of a mobile node have equal probability to be selected, i.e., $\frac{1}{V}$ and, thus, the selection is uniform and random. An increase in the value of alpha provides more weight to an unvisited waypoint with a smaller distance from the current waypoint. For *alpha* = ∞, an unvisited waypoint with the smallest distance (from the current waypoint) is selected.

Thus, we provide our waypoint selection scheme for steady-state SLAW for the following three cases:

- The waypoint selection scheme for *alpha*=0 (i.e., waypoint selection is random),
- The waypoint selection scheme for *alpha*=∞ (i.e., the closest waypoint is always visited first), and
- The waypoint selection scheme for 0 < *alpha* < ∞ (i.e., the next waypoint to be visited is chosen with probability inversely proportional to its distance from the mobile node's current location).

III. SLAW Without Pause

We first consider the scenario where a mobile node visits the selected set of waypoints with zero pause-time. In other words, a mobile node does not pause at any of the waypoints visited during simulation. In this work, we assume that the mobile node is traveling with a speed of 1 m/s (using 1 m/s is consistent with the node speed used in [7]). As discussed in Section II-A, a mobile node starts with a set of *x* clusters and a subset of waypoints (selected from each of these *x* clusters) to visit.

To start the SLAW mobility model in a steady-state, wechoose clusters and waypoints such that the selection of clusters and waypoints provides a steady-state distribution of mobile nodes over the network. Specifically, since a mobile node spends more time in bigger clusters than in smaller clusters, we choose a set of clusters w.r.t. the mean path length (MPL) for this set of clusters. Similarly, for a selected set of waypoints, the mobile node spends more time traveling between a pair of waypoints with larger distance between them and, thus, for this set of waypoints, our selection of waypoints is based on the path-length (PL) for this set of waypoints. Next, we define and discuss how we calculate MPL and PL for the three different cases defined in Section II-B (i.e., *alpha*=0, *alpha*=∞, and 0 < *alpha* < ∞).

A. SLAW with alpha = 0

For *alpha* = 0, selection of the next waypoint (to be visited) is completely random. Algorithm 2 provides a step-by-step description of the cluster and waypoint selection procedures that we follow to start SLAW in steady-state for α = 0.

Since in SLAW, a mobile node spends more time in a bigger cluster than in a smaller cluster, we use the average time spent in a cluster as a selection criteria. Specifically, if a mobile node chooses *x* clusters to visit, then to start in steady-state, the mobile node selects a set of x clusters with probability proportional to the mean-path length of the set of *x* clusters. For this purpose, we first calculate the mean path-length for every possible set of *x* clusters (**Step 2**). If C is the total number of clusters in the network, there are $S = \frac{C!}{x!(C-x)!}$ number of possible sets of *x* clusters. For every cluster-set *i* in *S*, we first estimate the mean path-length (MPL) for this cluster-set as:

Algorithm 2 Steady-State SLAW without pause: pseudocode

```
1.  for every possible set of x clusters do
2.     Calculate the mean path-length for this set of clusters.
3.  end for
4.  for each mobile node do
5.     while attempts < NUMBER do
6.        Select a set i of x clusters with probability ∝ MPL(i)
          (equation 4).
7.        Select y% of waypoints from every cluster in cluster-
          set i.
8.        Find the path-length (i.e., PL) of an ordered path of
          these selected waypoints.
9.     end while
10.    Select a set of waypoints j with probability ∝ PL(j).
11.    Uniformly place the mobile node on the path between
       two waypoints with probability ∝ the distance between
       the two waypoints.
12. end for
```

1) For every waypoint w in this cluster-set:
   a) Find the average of lengths of T paths formed by visiting all the waypoints (starting from waypoint w) in a random order. Specifically,

$$PL = \sum_{i=1}^{V-1} d_{k,k+1} \tag{2}$$

$$\overline{PL}_w = \frac{1}{T}\sum_{j=1}^{T} PL_j \tag{3}$$

$$MPL = \frac{1}{V}\sum_{w=1}^{V} \overline{PL}_w \tag{4}$$

where

- *V* is the total number of waypoints in a set of *x* clusters.
- $d_{k,k+1}$ is the distance between a successive pair of waypoints (i.e., *k* and *k* + 1) on an ordered path.
- PL is the length of an ordered path formed by visiting all the waypoints (starting from waypoint *w*) in a random order.
- $\overline{PL}_w$ is the average length of *T* paths formed by visiting all the waypoints (starting from waypoint *w*) in a random order.
- MPL is the mean path length for the set of *x* clusters.

If a cluster-set has *V* waypoints, there are a total of *V* ! possible paths formed by visiting all the waypoints ina random order. To save simulation time, we consider *T* =

25 paths to calculate the average of lengths of several paths formed per waypoint. We note that taking an average of $T = 25$ paths provides an accurate estimate of the average length of several paths per waypoint in a cluster-set. Once the MPL for every set of x clusters is calculated, a mobile node selects a set i of x clusters with probability proportional to MPL(i) (**Step 6**). The mobile node then chooses y% of waypoints from every cluster in this set (**Step 7**). For the selected waypoints, the path length (PL) of an ordered path (formed by visiting all the waypoints in a random order, starting from a randomly chosen waypoint) is calculated. For every mobile node, this process is repeated NUMBER = 300,000 times (Step 5). The node then selects a set j of waypoints with probability proportional to the path length of set j (i.e., PL(j); **Step 10**). The mobile node is placed over the ordered path formed for this set of waypoints. For this purpose, we first calculate the distance between every successive pair of waypoints on this ordered path. The mobile node is then placed between a pair of waypoints with probability proportional to the distance between the two waypoints (**Step 11**). Next, we describe our waypoint selection method used when a mobile node always selects the closest waypoint to visit first.

B. SLAW with alpha = ∞

When alpha $= \infty$ in the LATP algorithm, a mobile node always selects the closest waypoint (i.e., the waypoint with the minimum distance from the mobile node's current location) to visit. Since there is only one path possible if a mobile node visits the closest waypoint, MPL(i) for a cluster-set i of x clusters is calculated as:

1) For every waypoint w in cluster-set i:
   a) Find the length of a path formed by visiting all the waypoints (starting from waypoint w) such that the closest waypoint is always visited first. Specifically,

$$MPL = \frac{1}{V}\sum_{w=1}^{V} PL_w \tag{5}$$

where

- *MPL* is the average length of V paths (one per waypoint) formed by visiting all the waypoints in a set of x clusters.
- V is the total number of waypoints in cluster-set i.
- PL_w is the length of a path formed by visiting all the waypoints (starting from waypoint w) such that the mobile node always visits the closest waypoint.

Thus, for alpha = 1, we follow the Steps 1-12 listed in Algorithm 2; however, MPL is given by equation 5 and *PL* is calculated by forming an ordered path by visiting the closest waypoint first.

*C. SLAW with 0 < **alpha** < ∞*

As discussed in Section II-B, every unvisited waypoint is assigned a weight based on the value of *alpha* (equation 1). Next, we present simulation results for *alpha* = 0, *alpha* = ∞, and 0 < *alpha* < ∞.

D. Implementation Details (Simulation Set-up)

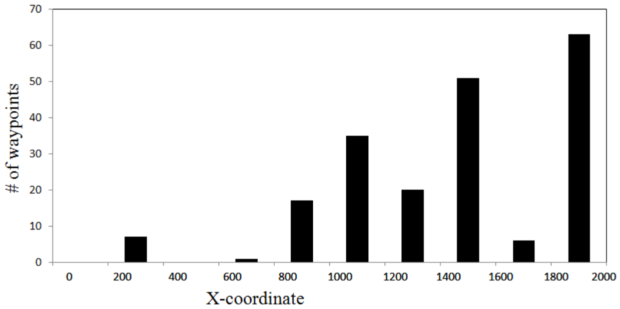
In this section, we provide details about our simulation setup and discuss the results. Table I lists the input parameters to SLAW and values used for simulations. Since a mobile node does not pause at any of the visited waypoints, values for β, min_p, max_p are listed as zero. (See Section II-A for details on SLAW input parameters.) As explained in Section III-A, we follow a sampling method for selecting an initial set of clusters and a set of waypoints. We call this sampling method SS SLAW. Thus, SS SLAW consists of the following steps:

- Select an ordered path formed by a set of waypoints for a mobile node and
- Place the mobile node on this path.

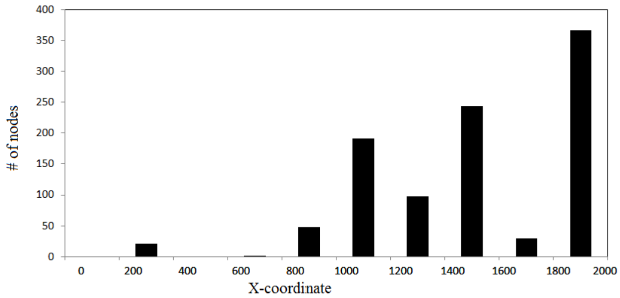
TABLE I. SLAW INPUT PARAMETERS AND THEIR VALUES USED FOR SIMULATIONS.

Input parameter	Value
$alpha$ (i.e., α)	0, 2, 3, ∞
$nodes$	1000
$area$ (m^2)	2000x2000, 10000x10000
wps	200, 1100, 10000
$hurst$	0.55, 0.75, 0.85
$range(m)$	100, 150, 200
$per\_waypoint$	10%
$clusters$	3
$beta$ (i.e., β)	0
min_pause	0
max_pause	0

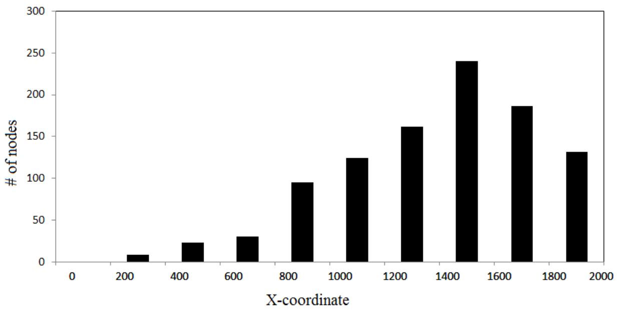
Specifically, we first find an ordered path formed by a set of waypoints visited in a random order and then place the mobile node between a pair of waypoints. We compare the distribution of mobile nodes in SLAW and SS SLAW by simulating several scenarios listed in Table I. To illustrate our sampling scheme used in SS SLAW, we consider a scenario with 1000 mobile nodes, 2000x2000 (m^2) simulation area, 200 waypoints, 0.55 hurst, and 100m transmission range. All the mobile nodes travel with a node speed of 1 m/s. As discussed in Section II-A, distribution of waypoints over the simulation area is modeled as a *self-similar* process. Thus, there is a certain amount of randomness involved in the distribution. In other words, using the same values for simulation *area*, *wps* (i.e., the number of waypoints), and the *hurst* parameter may lead to a different distribution of waypoints over the simulation area. Thus, the distribution of mobile nodes in SLAW and SS SLAW is compared using a fixed distribution of waypoints over the simulation area (generated via SLAW). We consider a fixed distribution of waypoints to make sure that the mobile nodes moving in SLAW and SS SLAW have the same distribution of waypoints to choose from. In this work, we consider a fixed distribution of waypoints for the scenario defined by (*area*, *hurst*, *range*, *wps*) = (2000x2000, 0.55, 100, 200). Specifically, Fig. 1(a) represents X-coordinates of the fixed distribution of waypoints for this scenario. For the fixed distribution of wps shown in Fig. 1(a), SLAW generates 20 clusters with a transmission *range* of 100*m*.



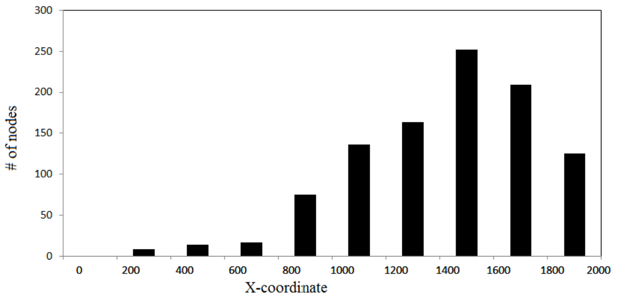
(a) A fixed distribution of X-coordinates for waypoints over the simulation area.



(b) Distribution of X-coordinates for 1000 mobile nodes in SLAWat 0 seconds.



(c) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SLAW at 10th hour.



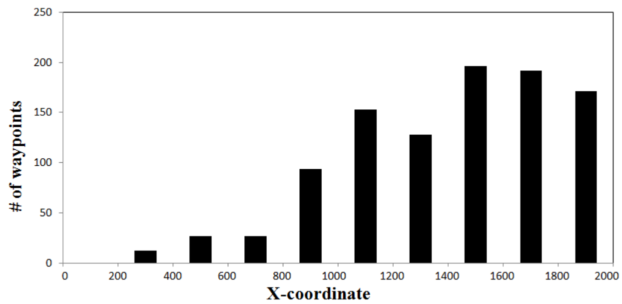
(d) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SS SLAW at 0 seconds.

Fig. 1. Distribution of X-coordinates for 200 waypoints and 1000 mobile nodes in SLAW and SS SLAW without pause and for alpha = 0 (i.e., random selection of waypoints).

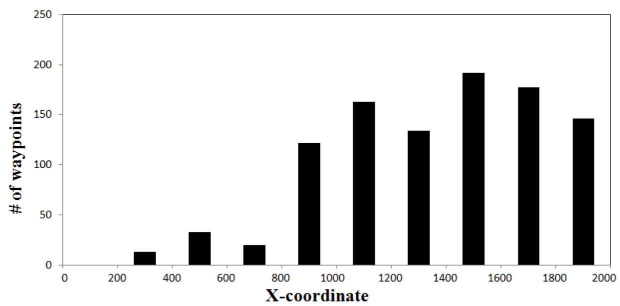
To find the steady-state distribution of mobile nodes in SLAW, we simulated SLAW for several hours. The simulation-time required for SLAW to come to a steadystate depends upon the type of scenario. For example, for a scenario defined by (*area*, *hurst*, *range*, *wps*) = (2000x2000, 0.55, 100, 200) SLAW comes to a steadystate quickly, maximum of 10 hours. However, for a scenario defined by (10000x10000, 0.85, 150, 10000), SLAW takes more than 1000 hours to come to steady-state. We adopt the following approach to estimate the simulationtime required before SLAW reaches its steady-state:

1) Run SLAW for several hours.
2) Find the distribution of *X* and *Y* coordinates of mobile nodes at N time intervals.
3) To determine if SLAW has reached its steady-state at time instant T_l, perform Chi-Square test between the *X*(*Y*) distribution of mobile nodes obtained at time T_l and the *X*(*Y*) distribution obtained at every successive time instant (i.e., $T_{l+1}, T_{l+2}, ...T_N$).
   a) If the Chi-Square test shows that the difference among the distributions analyzed is nonsignificant
   then
      i) SLAW has reached its steady-state at time instant T_l
      else
      i) Repeat Steps above for time instant T_{l+1} and $T_{l+2}, ...T_N$).

For the scenario defined by (2000x2000, 0.55, 100, 200), we simulated SLAW for 100 hours. Table II lists the results for the Chi-Square test performed among distributions obtained at the 10th hour and every 10 hours. Since there are no waypoints to choose from in the 0-200 interval, the degrees of freedom for the Chi-Square test results presented in this work is 8. As shown in the table, the differences among the distributions are non-significant, which suggests that SLAW is near its steady-state distribution at the 10th hour.



(a) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SLAW at 10th hour.

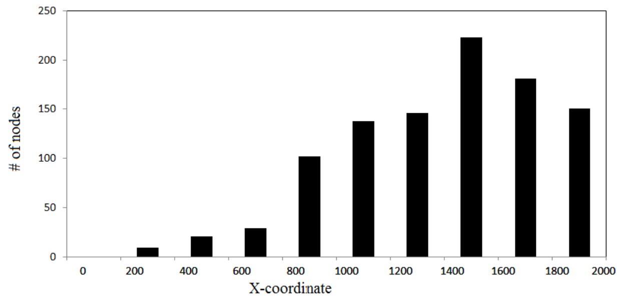


(b) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SS SLAW at 0 seconds

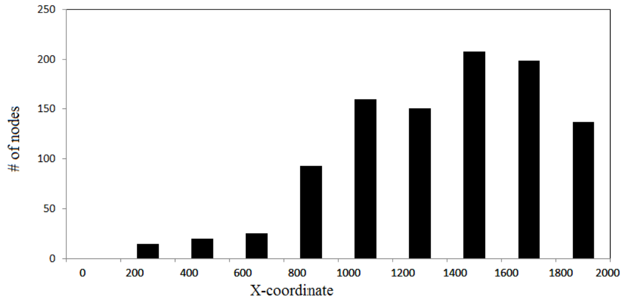
Fig. 2. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW without pause and for *alpha* = ∞ (i.e., closest waypoint is selected).

Fig. 1(c) represents the X-coordinates of 1000 mobile nodes at the 10th hour in SLAW for *alpha* = 0. Since SS SLAW starts SLAW in its steady-state, we plot the distribution of X-coordinates of mobile nodes in SS

SLAW at 0 seconds (i.e., at the start of the simulation). Fig. 1(d) shows the distribution of mobile nodes in SS SLAW at 0 seconds for *alpha* = 0. Similarly, to compare SLAW and SS SLAW for *alpha* = ∞, we simulate SLAW and SS SLAW such that while moving, a mobile node always selects the closest waypoint first. Fig. 2 represents the X-coordinates for 1000 mobile nodes in SLAW and SS SLAW for *alpha* = ∞ at the 10th hour and 0 seconds, respectively.

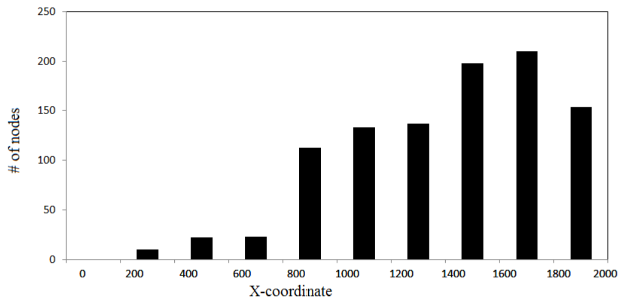


(a) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SLAW at 10th hour.

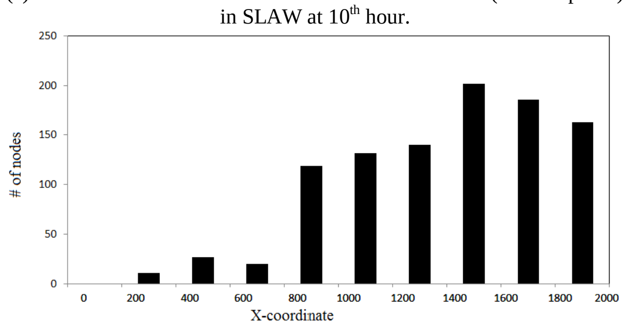


(b) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SS SLAW at 0 seconds.

Fig. 3. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW without pause and for *alpha* = 2.



(a) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SLAW at 10th hour.



(b) Distribution of X-coordinates for 1000 mobile nodes (without pause) in SS SLAW at 0 seconds.

Fig. 4. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW without pause and for *alpha* = 3.

An analysis of the real GPS traces performed by the authors of SLAW suggests that the *alpha* values for the GPS traces [7] vary between 1 and 3. Thus, for validating our sampling technique for values other than 0 and ∞, we simulated SLAW and SS SLAW for alpha = 2 and *alpha* = 3. Fig. 3 and Fig. 4 compare the distributions obtained by simulating SLAW and SS SLAW for *alpha* = 2 and *alpha* = 3, respectively. To compare these distributions statistically, we perform another Chi-Square test. Table IV shows the results for the Chi-Square test performed on distributions obtained in Fig. 1, Fig. 2, and Fig. 3. For example, Scenario "Random No pause" means the scenario with (*alpha* = 0, without pause) and, thus, refers to a comparison between distributions represented by Fig. 1(c) and Fig. 1(d). Since there are no waypoints to choose from 0-200 interval, degree of freedom for the Chi-Square test is 8. The results obtained from Chi-Square test suggest that our sampling technique approximates the stationary distribution of SLAW at the start of the simulation.

TABLE II. RESULTS FOR THE CHI-SQUARE TEST PERFORMED ON THE DISTRIBUTION OBTAINED BY SLAW AT 10TH HOUR AND THE DISTRIBUTIONS OBTAINED AT EVERY 10 HOURS.

Hours	Chi-Sq	P-Value
20	3.436	0.904
30	3.934	0.863
40	5.951	0.653
50	13.610	0.093
60	17.438	0.026
70	8.842	0.356
80	5.585	0.694
90	10.337	0.242
100	8.444	0.391

IV. SLAW WITH PAUSE

In this section, we consider the scenario where a mobile node pauses at every visited waypoint. Specifically, a mobile node travels from one waypoint to another and after it reaches the destination waypoint, it pauses for a pause-time. Several studies [7], [9] have suggested that the *pause-time* distribution for mobile nodes follows a truncated power-law. The authors of SLAW validated that the ICTs distribution of mobile nodes in their real GPS traces fit a truncated Pareto distribution. Specifically, the density and distribution functions for a truncated Pareto random variable, p, are given by [12]:

$$z(p) = \frac{\beta * p_{min}^{\beta} * p^{-\beta-1}}{1 - \left(\frac{p_{min}}{p_{max}}\right)^{\beta}}, \quad p_{min} \le p \le p_{max} \quad (6)$$

$$Z(p) = \begin{cases} 0 & \text{p} < p_{min} \\ \frac{1 - p_{min}^{\beta} * p^{-\beta}}{1 - \left(\frac{p_{min}}{p_{max}}\right)^{\beta}} & p_{min} \le p \le p_{max} \\ 1 & \text{p} > p_{max} \end{cases} \quad (7)$$

where

- $\beta > 0$ and describes the asymptotic behavior of the distribution.
- $p_{min} > 0$ and is the minimum pause value for a mobile node, and
- $p_{max} > p_{min}$ and is the maximum pause value for a mobile node.

When a mobile node moves from one waypoint to another (i.e., one flight), its flight-time depends on its speed and the distance between the two waypoints. Once the mobile node reaches its destination waypoint, it pauses there for a finite amount of time, (i.e., *pause-time*). Thus, the fraction of time a mobile node remains paused during the simulation depends upon the average amount of time it spends in a paused state and the average amount of time it spends traveling from one waypoint to another. To calculate the probability that a mobile node is paused, we estimate the average amount of time it spends pausing at a waypoint and the average amount of time it spends traveling from one waypoint to another. Specifically,

$$Pr(pause) = \frac{E[P]}{E[P] + E[T]} \tag{8}$$

where

- *Pr*(*pause*) is the probability that a mobile node is paused,
- *E*[*P*] is the mean value for the *pause-time* (i.e., *P*) of a mobile node, and
- *E*[*T*] is the mean value for the *travel-time* (i.e., *T*) of a mobile node.

Since *pause-time*, P, for a mobile node is truncated Pareto distributed (equations 6-7), the expected value of *P* is given by:

$$E[P] = \frac{p_{min}^{\beta}}{1 - \left(\frac{p_{min}}{p_{max}}\right)^{\beta}} . \frac{\beta}{\beta - 1} . \left. \frac{1}{p_{min}^{\beta-1}} - \frac{1}{p_{max}^{\beta-1}} \right), \quad \beta > 1 \tag{9}$$

In this work, we assume that the mobile node is moving with a constant speed of 1 m/s. Thus, with a constant node speed of 1 m/s, *E*[*T*] (in seconds) is given by:

$$E[T] = E[D] \tag{10}$$

where *D* is the average distance between two waypoints (in meters). To obtain *E*[*D*], we first calculate the distance between every pair of waypoints selected for a mobile node and average. Specifically,

$$E[D] = \frac{\sum_{i=1}^{V} \sum_{j=i+1}^{V} d_{ij}}{V(V-1)/2} \tag{11}$$

where

- d_{ij} is the distance between waypoints i and j and
- *V* is the total number of waypoints selected for a mobile node.

Using equations 9 and 11, *Pr*(*pause*) is calculated (equation 8).

Algorithm 3 Steady-State SLAW with pause: pseudocode

```
1.  for each mobile node do
2.    while attempts < NUMBER do
3.      Select x clusters randomly from the network.
4.      Select y% of waypoints from each of the selected x clusters.
5.      Find path-length (i.e., PL) of an ordered path of these selected waypoints.
6.    end while
7.    Select a set of waypoints j with probability ∝ PL(j).
8.    Calculate Pr(pause) and generate a random number U uniformly in (0,1).
9.    if (U < Pr(Pause)) then
10.     The mobile node is paused at a randomly selected waypoint.
11.   else
12.     The mobile node is placed between two waypoints with probability proportional to the distance between the
        waypoints the pair of waypoints.
13.   end if
14. end for
```

A. SLAW with α = 0

In this section, we describe our cluster and waypoint selection scheme for α = 0 (i.e., the selection of the next waypoint to be visited by a mobile node is completely random). Algorithm 3 describes the steps that we follow for starting SLAW (with pause) in a steady-state. Further details on the algorithm follow. Since a mobile node pauses at each of the selected waypoints, on average it spends *E*[*P*] amount of time pausing at a waypoint. Thus, in this case, our cluster selection scheme is not based on MPL; instead, it is completely random (**Step 3**). Once clusters are selected, a mobile node selects y% of waypoints from each of the selected *x* clusters (**Step 4**). To estimate the amount of time a mobile node spends visiting the selected waypoints, we calculate the length of a path formed by visiting these waypoints in a random order. Specifically, starting from one waypoint (selected uniformly from the set of selected waypoints), an ordered path is formed by visiting waypoints randomly (**Step 5**). As discussed in Section III-A, for *V* waypoints in a cluster-set, there are a total of *V*! paths formed by visiting all the waypoints in a random order. To save simulation time, we consider *T*=25 paths to calculate the average length of paths per waypoint in a clusterset. For every mobile node, NUMBER=300,000 different sets of waypoints are selected (**Step 2**) and the node then selects a set of waypoints *j* with probability proportional to path length of set *j* (Step 7). Since the mobile node is paused at every waypoint it visits, path-length (i.e., *PL*) for every set of waypoints is estimated in Step 4 as follows:

$$PL = \sum_{i=1}^{V} d_{k,k+1} + P_{k+1} \tag{12}$$

where

- *PL* is the length of an ordered path formed by visiting the set of waypoints (chosen from clusterset *i*) in random order by a mobile node.
- *V* is the total number of waypoints chosen from cluster-set *i* by a mobile node.
- *P* is the amount of time for which a mobile node pauses at a waypoint.

Once path lengths for all sets of waypoints are calculated, a mobile node selects a set of waypoints, i, with probability proportional to *PL*(*i*) (**Step 7**). To determine if the mobile node begins in a moving state or in a pausedstate, we calculate *Pr*(*pause*) and uniformly generate a random number, *U*, between (0,1) (**Step 8**). If *U* < Pr (pause), the mobile node begins in a paused-state at a uniformly selected waypoint at random (**Step 10**). Otherwise, the mobile node begins in a moving state and is placed between a pair of waypoints with probability proportional to the distance between the two waypoints (**Step 12**). If the mobile node begins in a paused state, we calculate its initial *pause-time* via a fundamental result from renewal theory [5]. Specifically, the CDF of the mobile node's initial *pause-time* is given by:

$$Z_0(p) = \frac{\int_0^p [1 - Z(t)]dt}{E[P]} \tag{13}$$

where

- $Z_0(p)$ is *the* CDF of P_0 (i.e., initial pause-time),
- *Z*(*p*) is *the* CDF of pause-time, P (equation 7), and
- *E*[*P*] is *the* mean value of P (equation 9).

Using values of *Z*(*t*) and *E*[*P*], equation 13 yields:

$$Z_0(p) = \quad \frac{max_p^{-\beta} * x(1-\beta) - x^{1-\beta}}{\beta(p_{min}^{1-\beta} - p_{max}^{1-\beta})}\Bigg), \beta > 1 \tag{14}$$

Thus, we use the CDF $Z_0(p)$ to sample the value for the initial pause-time, P_0. Every successive *pause-time* value for a mobile node is thus sampled using the CDF given by equation 7. Next, we describe our waypoint selection method for SLAW (with pause) when a mobile node always visits the closest waypoint first.

B. SLAW with $\alpha = \infty$

As discussed in Section IV-A, there is only one path when a mobile node visits the closest waypoint first. Also, since the node pauses at every waypoint that it visits, for starting SLAW in steady-state, our choice of clusters is random. Thus, for $\alpha = \infty$, we follow Steps 1-12 from Algorithm 3, except that the *PL* is calculated for an ordered path such that the closest waypoint is always visited first.

C. SLAW with 0 < alpha < 1

As discussed in Section III-C, for 0 < *alpha* < ∞, probability of choosing an unvisited waypoint is given by equation 1. To validate our sampling scheme for values other than 0 and ∞, we simulate both SLAW and SS SLAW (with pause) for *alpha* = 2 and *alpha* = 3.

D. Implementation Details (Simulation Set-up)

In this section, we provide details about the simulation setup and discuss the results. Since mobile nodes pause at every waypoint visited, Table III lists the values of input parameters that define the power-law distribution used in this work. (See Section II-A for details of input parameters to SLAW.)

TABLE III. SLAW INPUT PARAMETERS FOR PAUSE-TIME DISTRIBUTION AND THEIR VALUES USED IN SIMULATIONS.

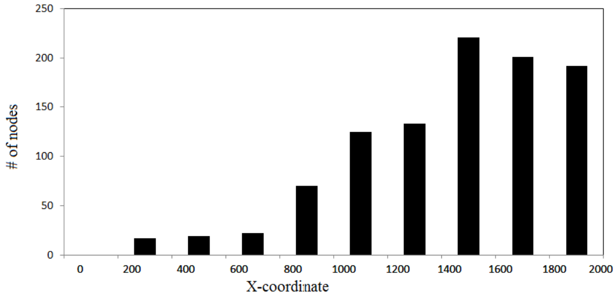
Input parameter	Value
beta (i.e., β)	1.5
min_pause	30 (sec)
max_pause)	36000 (sec)

As explained in Section IV-A, to start the SLAW mobility model (with pause) in a steady-state, our sampling scheme, SS SLAW, involves the following steps:

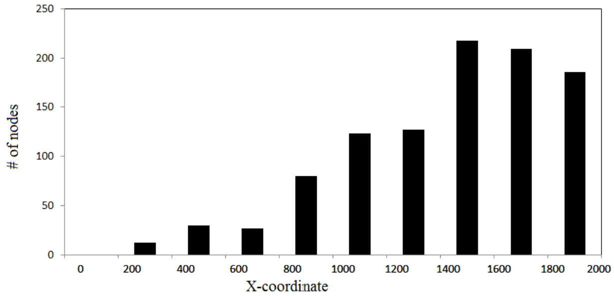
- Select an ordered path formed by a set of waypoints for a mobile node,
- Calculate the probability of pause,
- If the node is paused, place it on a randomly selected waypoint,
- Else, place the mobile node uniformly on the selected path.

To compare the distribution of mobile nodes in SLAW and SS SLAW, we simulated 1000 mobile nodes in a 2000x2000 (m^2) area. (See Section I for further details on simulation parameters and setup.) Fig. 5(a) represents the X-coordinates of 1000 mobile nodes at the 1000^{th} hour in SLAW for *alpha* = 0. Fig. 5(b) shows the distribution of mobile nodes in SS SLAW at 0 seconds for *alpha* = 0. Similarly, we compare the distributions obtained from SLAW and SS SLAW for $\alpha = \infty$. Fig. 6 represents the X-coordinates for 1000 mobile nodes in SLAW and SS SLAW for *alpha* = ∞ at 1000^{th} hour and 0 seconds, respectively. To validate SS SLAW for other values of alpha, we simulated SLAW and SS SLAW for *alpha* = 2 and *alpha* = 3. Fig. 7 and Fig. 8 compare the distributions of mobile nodes obtained from SLAW and SS SLAW for *alpha* = 2 and *alpha* = 3, respectively.

Fig. 5, Fig. 6, Fig. 7, and Fig. 8 provide visual evidence that SS SLAW provides a very close approximation for distributions of mobile nodes at steady-state in SLAW. To illustrate further, we provide results for another Chi- Square test performed on these sets of distributions. Table IV provides Chi-Square values and the corresponding Pvalues for the various distributions plotted in Figures 1-8. The results of the Chi-Square test suggest that the statistical differences among distributions obtained from SLAW (at steady-state) and the distributions obtained from SS SLAW are non-significant.

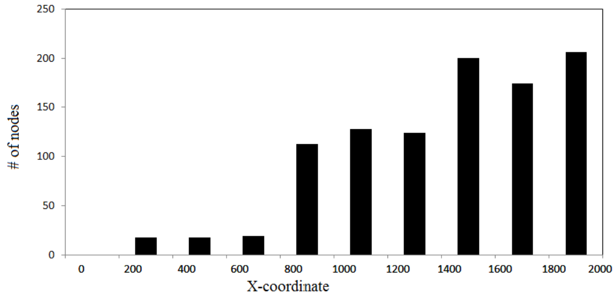


(a) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SLAW at 10th hour.

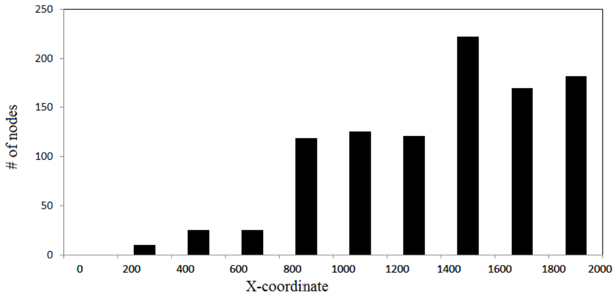


(b) Distribution of X-coordinates for 1000 mobile nodes (with pause) SS SLAW at 0 seconds.

Fig. 5. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW with pause and for α = 0 (i.e., random selection of waypoints).

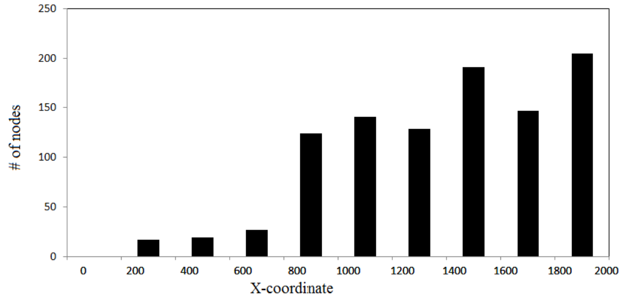


(a) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SLAW at 10th hour.

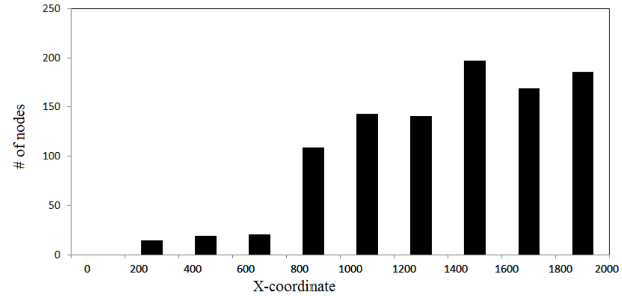


(b) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SS SLAW at 0 seconds.

Fig. 6. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW with pause and for α = ∞ (i.e., closest waypoint is selected).

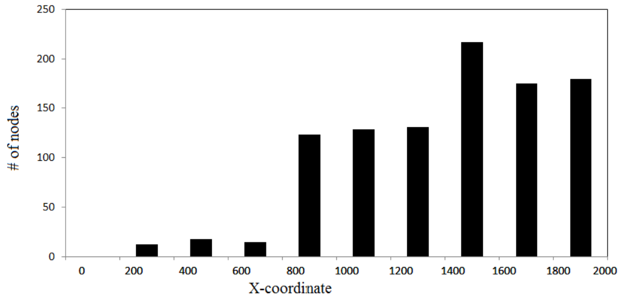


(a) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SLAW at 10th hour.

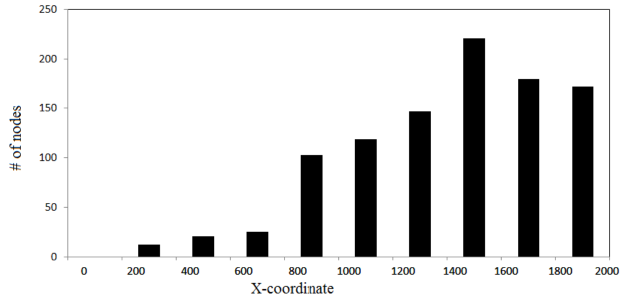


(b) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SS SLAW at 0 seconds.

Fig. 7. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW with pause and for α = 2.



(a) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SLAW at 10th hour.



(b) Distribution of X-coordinates for 1000 mobile nodes (with pause) in SS SLAW at 0 seconds.

Fig. 8. Distribution of X-coordinates for 1000 mobile nodes in SLAW and SS SLAW with pause and for α = 3.

TABLE IV. RESULTS FOR THE CHI-SQUARE TEST ON VARIOUS DISTRIBUTIONS OBTAINED FROM SLAW AND SS SLAW.

Scenario	Chi-Sq	P-Value
Random_No_Pause	10.527	0.230
Random_With_Pause	1.474	0.993
Closest_No_Pause	8.389	0.396
Closest_With_Pause	8.456	0.390
Alpha=2_No_Pause	5.968	0.651
Alpha=2_With_Pause	4.936	0.764
Alpha=3_No_Pause	5.514	0.7015
Alpha=3_With_Pause	6.114	0.635

V. CONCLUSION

In this work, we have developed steady-state distributions for the SLAW mobility model, a recently published mobility model based on real GPS traces. While it is important to use a mobility model whose synthetic traces closely match the real GPS traces, it is critical to start the mobility model in a steady-state. Starting a mobility model in its steady-state eliminates the need for inefficient traditional methods of avoiding any initialization bias in data. Since, in SLAW, the movement patterns of mobile nodes is affected by an input parameter called *alpha*, we have provided steady-state distributions for different scenarios created using different values of *alpha*.

REFERENCES

[1] A. Munjal, T. Camp, and N. Aschenbruck, "Changing trends in modeling mobility: A simple way to model human walks," *Journal of Electrical and Computer Engineering*, vol. 2012, pp. 16, 2012.

[2] [Online]. Available: http://crawdad.cs.dartmouth.edu/index.php.

[3] N. Aschenbruck, A. Munjal, and T. Camp, "Trace-based mobility modeling for multi-hop wireless networks," *Computer Communications*, pp. 704-714, 2011.

[4] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Comm. & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, vol. 2, no. 5, pp. 483-502, 2001.

[5] W. C. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transaction on Mobile Computing*, vol. 3, no. 1, Jan-March 2004.

[6] F. Bai and A. Helmy. A survey of mobility models. [Online]. Available: http://www.cise.ufl.edu/˜helmy/papers/Survey-Mobility-Chapter-1.pdf

[7] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "SLAW: A mobility model for human walks," in *Proc. INFOCOM*, 2009.

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

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

[10] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. IEEE/ACM Transactions on Networking*, 1994, pp. 1-15.

[11] R. G. Clegg, "A practical guide to measuring the Hurst parameter," in *Proc. 21st UK Performance Engineering Workshop, School of Computing Science Technical Report Series*, CSTR-916, University of Newcastle, 2006.

[12] WIKIPEDIA: Pareto. [Online]. Available: http://en.wikipedia.org/wiki/Pareto_

[13] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Demystifying levy walk patterns in human walks," Techical Report, NCSU, 2008.

[14] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proc. MobiCom*, 2003.

[15] J. Yoon, B. Noble, M. Kiu, and M. Kim, "Building realistic mobility models from coarse-grained traces," in *Proc. MobiSys*, 2006.

[16] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proc. 26th IEEE International Conference on Computer Communications*, 2007, pp. 758-766.

[17] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," in *Proc. INFOCOM*, 2006.

[18] A. Mei and J. Stefa, "SWIM: A simple model to generate small mobile worlds," in *Proc. INFOCOM*, 2009, pp. 2106-2113.

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

Aarti Munjal received her Ph.D. (CS) from Colorado School of Mines in 2012, an M.Tech (ICT) degree from Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT) in 2007, and a B.Tech (CE) degree from Kurukshetra University in 2003. She is currently an Assistant Research Professor in the Department of Biostatistics and Informatics at the Colorado School of Public Health, University of Colorado Denver. Her research interests include modeling simulations for mobile networks and using statistical methods/analysis to estimate power and sample-size for biomedical studies.

William C. Navidi received the BA degree in mathematics from New College, the MA degree in mathematics from Michigan State University, and the MA and PhD degrees in statistics from the University of California at Berkeley. He is currently a Professor in the Department of Applied Mathematics and Statistics at the Colorado School of Mines. His research interests include the development and analysis of broadcast protocols for mobile ad hoc networks. Dr. Navidi is a member of IMS, ASA, and WNAR.

Tracy Camp received the BA degree in mathematics from Kalamazoo College in 1987, the MS degree in computer science from Michigan State University in 1989, and the PhD degree in computer science from The College of William and Mary in 1993. Currently she is a Professor in the Department of Electrical Engineering and Computer Science at the Colorado School of Mines. Her main field of research is in the area of mobile computing and networking. She is a member of IEEE, ACM, and ACM-W.