

# Perfect Simulations for Random Trip Mobility Models

Santashil PalChaudhuri  
Computer Science, Rice University  
Houston, Texas, USA  
santa@cs.rice.edu

Jean-Yves Le Boudec  
EPFL  
Lausanne, Switzerland  
jean-yves.leboudec@epfl.ch

Milan Vojnović  
Microsoft Research  
Cambridge, UK  
milanv@microsoft.com

## Abstract

*The random trip model was recently proposed as a generic mobility model that contains many particular mobility models, including the widely-known random waypoint and random walks, and accommodates more realistic scenarios. The probability distribution of the movement of a mobile in all these models typically varies with time and converges to a “steady state” distribution (viz. stationary distribution), whenever the last exists. Protocol performance during this transient phase and in steady-state may differ significantly. This justifies the interest in perfect sampling of the initial node mobility state, so that the simulation of the node mobility is perfect, i.e. it is in steady state throughout a simulation. In this work, we describe implementation of the perfect sampling for some random trip models. Our tool produces a perfect sample of the node mobility state, which is then used as input to the widely-used ns-2 network simulator. We further show some simulation results for a particular random trip mobility model, based on a real-world road map. The performance metrics that we consider include various node communication properties and their evolution with time. The results demonstrate difference between transient and steady-state phases and that the transient phase can be long lasting (in the order of a typical simulation duration), if the initial state is drawn from a non steady-state distribution. The results give strong arguments in favor to running perfect simulations. Our perfect sampling tool is available to public at: <http://www.cs.rice.edu/~santa/research/mobility>.*

## 1 Introduction

The most common way to study mobile ad hoc networks is through simulations. Even though real world deployment is essential to understand the effectiveness and performance of ad hoc networks, simulations have some advantages. Simulations are fast and repeatable, and it is possible in simulators to isolate parameters affecting the performance of a design. It also allows testing of a wide variety of scenarios, along with large ones, which is difficult if not

impossible in a real network.

Therefore a variety of mobility models and communication patterns have been developed in the simulators for performance evaluation of a design. It is important to use realistic mobility models so that the evaluation results will have a close correlation to the performance when actually deployed. However, the most commonly used mobility model is the Random Waypoint Mobility (RWM) [1], which though being simple and easy to simulate is not very realistic. Recently though, various mobility models have been proposed which attempt to better capture mobility traces than RWM. These mobility models range from Obstacle Mobility Model [2] to City Section Mobility models (see the survey [3]).

All of the proposed models suffer from non steady-state distribution at the start of a simulation. The probability distribution of the movement of nodes typically varies with time, and ultimately converges to a steady-state distribution, known in probability literature as *stationary distribution*. This convergence time varies widely, depending on the parameters of the mobility model and is not deterministic. The performance of the network varies with time, and there might be a substantial difference between startup time and after steady state has been reached. This was first noticed by Yoon, Liu, and Noble [4] for RWM models. The suggested method to deal with this problem (in the rare cases when it is addressed at all) was to discard the initial set of observations hoping that steady state would have set in. Other than throwing away useful computation work, it is extremely difficult to predetermine the length of this transient period. Navidi and Camp [5] recently solved the problem for RWM models by finding the steady state distribution, and initializing the mobility state to a sample drawn from the steady state distribution (this is classically called “perfect simulation” and the sampling “perfect sampling”). But as explained earlier, this RWM model is not realistic for the evaluation of a protocol design.

Le Boudec and Vojnović [6] recently used Palm calculus to provide a generic framework for finding the steady state distribution of the mobility state for a large class of random mobility movements – they call, random trip models. In this work, we derive the steady state distributions for some ran-

dom trip mobility models and implement perfect sampling in a tool, whose output is directly used by *de-facto* standard *ns-2* network simulator. In particular we implemented the RWM, random walk with wrapping, random walk with reflection, restricted random waypoint and its special case space graph. All these models are instances of the random trip model. Our simulation evaluation is for a space graph model. The results show considerable difference of the mobility characteristics in initial and steady state and its resulting effect on metrics of interest for routing performance.

Why do we want to run mobility in a steady-state? It is legitimate to argue that in some situations, one may be interested in performance of some protocols under a transient mobility. One argument in favor to steady-state mobility simulations is as follows. We are commonly interested in average-case performance of a protocol which is captured by long-run averages. If the node mobility ultimately converges to a steady-state, then asymptotically, the long-run averages are determined by the mobility steady-state.

Some mobility models are specified by real-world empirical mobility traces. Under hypothesis that the trace is a realization of a stationary process, no perfect simulation is needed. The hypothesis is true if we assume a node mobility is a stationary realization, and we trace its movement from an arbitrary time. There have been studies of protocols performance by using real-world movement traces, for example that using a fleet of city buses in Seattle metropolitan area [7]. In order to instantiate a realistic sample of a space graph, we use existing real-world city maps. United States Census Bureau makes available detailed street maps for the whole of United States, based on the bureau’s TIGER (Topologically Integrated Geographic Encoding and Referencing) database [8]. One may even further make our simulations more realistic, by specifying the trip selection rule (introduced later) so as to match empirical frequency counts of trips and their properties (e.g. trip duration, trip origin and destination, trip mode, etc), which are available from various transportation systems survey studies. However, we do not pursue this direction here.

The paper is organized as follows. In Section 2, we give a brief background on how Palm calculus has been applied to derive the generic framework of steady-state mobility models and present perfect sampling algorithm for restricted random waypoint, which accommodates our example space graph mobility model. Section 3 discusses our implementation of the perfect sampling. Section 4 shows simulation results, obtained for a particular space graph. We conclude in Section 5.

## 2 Random Trip Model and its Perfect Simulation

In this section we first recall the definition of random trip, a class of mobility models defined in [6], and its perfect simulation.

### 2.1 Random trip

The random trip is a generic mobility model for individual movements. It is defined by a set of paths over a connected domain  $\mathcal{A}$ , an initialization rule, and a trip selection rule. A path is defined as a mapping from  $[0, 1]$  to  $\mathcal{A}$ . For example, we use later in this paper a special case called space graph where  $\mathcal{A}$  is a graph representing streets in a city; the set of paths is made of all shortest paths between any two intersections in the city (represented by vertices in the graph). Another example is city-section, which differs from the space graph in that the trip endpoints are not restricted to vertices of the graph, but take values on the whole domain  $\mathcal{A}$ . A trip is the combination of a path and a duration, or, equivalently, a path and a numerical speed factor at which the path is run. Initially, a mobile picks a path from the set according to the model’s initialization rule, and goes along the path. At the end of the trip, the mobile picks another path according to the model’s trip selection rule. The newly chosen path is restricted to start where the previous ended. For example, the initialization rule we use in the space graph example consists in picking a location uniformly on the set of street intersections; the trip selection chooses as next trip a pause at the current location if the finishing trip is not a pause; after a pause, the trip selection rule chooses a new endpoint uniformly on the graph vertices and a numerical speed in some specified interval  $[v_{\min}, v_{\max}]$ .

The random trip model is quite general. It contains as special cases: the classical random waypoint, random walks with reflection or wrapping, “fish in a bowl”, restricted random waypoint, and others. In this paper we focus the formalism on “restricted random waypoint”, which contains as special case the space graph example. With restricted random waypoint, we have a given collection of sub-domains  $\mathcal{A}_\ell$  that are all contained in  $\mathcal{A}$ . The mobility of a node is then contained in  $\mathcal{A}$ , with trip end points in sub-domains  $\mathcal{A}_\ell$ . With space graph, we define only one sub-domain,  $\mathcal{A}_1$ , as the set of the vertices of the space graph. In this paper, we confine our formalism to restricted random waypoint with only one sub-domain. The general case bears some notational complexity; see [6] for definition, properties, and perfect sampling.

Formally, we consider a mobile node that lives in a connected set  $\mathcal{A}$  (a subset of  $\mathbb{R}^n$  for  $n = 1$  or  $n = 2$  or  $n = 3$ ). The trip-end times are  $T_0 = 0 < T_1 < T_2 < \dots$ . Let  $S_n := T_{n+1} - T_n$  be the duration of the  $n$ -th trip. At a trip transition instant  $T_n$ , the node selects the path  $P_n : [0, 1] \rightarrow \mathcal{A}$  of the  $n$ -th trip. The mobile position at time  $t$  is given by

$$X(t) = P_n \left( \frac{t - T_n}{S_n} \right), \quad T_n \leq t < T_{n+1}.$$

In particular, the node position at the trip begin-point is  $X(T_n) = P_n(0)$  and at the trip end-point,  $X(T_{n+1}) = P_n(1)$  (we assume that  $(P_n)_n$  is such that  $P_n(1) =$

$P_{n+1}(0)$ , for all  $n$ ). We denote with  $U(t) = (t - T_n)/S_n$  the fraction of the elapsed time on the  $n$ -th trip. Thus, we can write  $X(t) = P(t)U(t)$ , where  $P(t) = P_n$ ,  $T_n \leq t < T_{n+1}$ .

We assume that the trip selection rule is Markov modulated, i.e. there is a sequence of phases  $(I_n)_n$  that take values on some state space  $\mathcal{I}$ . The trip selection rule specifies that at a transition instant  $T_n$ , the path  $P_n$  and trip duration  $S_n$ , given the phase  $I_n$  and the mobile position  $M_n := X(T_n)$ , are drawn independently of  $n$  and any other past. In simple cases, the phase corresponds to being in a pause or move state; in more elaborate cases, the phase can be used to model inter-city and intra-city trips.

The original definition of random trip models [6] uses additional assumptions. The phases  $(I_n)_n$  are assumed to be a Markov chain. The node position  $M_{n+1}$  (at the end-point of the  $n$ -th trip,  $T_{n+1}$ ) is assumed to be conditionally independent of all past phases, but the phase  $I_n$ .

The mobility state at time  $t$  is  $Z(t) := (I(t), P(t), S(t), U(t))$ . Under additional technical assumptions in [6],  $Z(t)$  is sufficient to entirely describe the future evolution of the mobility state. Further, any random trip mobility model has a unique time-stationary distribution if and only if the mean trip duration averaged at trip transition instants is finite (Theorem 3 [6]). For the city section example, this means that  $v_{\min} > 0$  and the average pause time is finite [6].

## 2.2 Perfect Simulation

As explained in the introduction (and illustrated in the rest of the paper), the random trip model may take a long time to converge. It is possible to directly sample from the stationary distribution of  $Z(t)$  (when the condition for its existence is satisfied), thus achieving a “perfect” simulation of the mobility model, namely one that is free of transients. This is achieved in [6] by using Palm calculus, a set of formulae that relate the distributions of a system at an arbitrary point in time and at an arbitrary transition instant (here the trip-end times) – see for example [9] for more explanations. Note that those two distributions are different: if we sample a mobile speed in the city graph at an arbitrary trip-end time (this is called the “Palm” distribution), we obtain a uniform distribution over  $[v_{\min}, v_{\max}]$ ; in contrast, the mobile speed sampled at an arbitrary point in time is more likely to be small, since a mobile spends more time, in average, on trips that have a small speed. Similarly, the Palm distribution of a mobile position is uniform over the graph, but the mobile position as sampled at an arbitrary point is more likely to be on an edge of the graph that lie on a trip path with a large average trip duration.

The perfect sampling algorithm is described in Figures 1 and 2 in the case with one sub-domain  $\mathcal{A}_1$  (the general case has some notation complexity that obscures the main ideas). We implemented the general case in the tool that we describe in this paper.

```

Draw phase  $\phi$ , previous and next trip endpoints
 $M_0$  and  $M_1$  by the sampling algorithm in Figure 2
if  $\phi = move$ 
  Sample a speed  $v$  from density proportional to  $\frac{1}{v} f_V^0(v)$ 
  Draw  $u \sim \text{Unif}(0, 1)$ , set the current mobile position
  to  $uM_0 + (1 - u)M_1$ 
  Set numeric speed to  $v$  and set end of trip to point  $M_1$ 
else
  Sample  $t$  from density defined  $f_T^0(t) = \frac{1}{T} \int_t^\infty f_S^0(s) ds$ 
  Set the mobile position to  $M \sim \text{Unif}(\mathcal{A}_1)$ 
  Schedule end of pause at time  $t$ 
end

```

**Figure 1. Perfect sampling for restricted random waypoint with one sub-domain  $\mathcal{A}_1$ . Notation:  $f_V^0(t)$  is the density of the next numerical speed chosen by the trip selection rule (for example uniform in  $[v_{\min}, v_{\max}]$ );  $f_S^0(t)$  is the density of the pause time chosen by the trip selection rule (for example uniform in  $[0, T_{\max}]$ ). For example, with space graph  $\mathcal{A}_1$  is the set of vertices; with city-section  $\mathcal{A}_1 = \mathcal{A}$ .**

```

Let  $q_0 = \tau_{pause} / (\tau_{pause} + \omega \Delta)$ 
do forever
  Draw  $U_1 \sim \text{Unif}(0, 1)$ 
  if  $(U_1 \leq q_0)$   $\phi = pause$ ; leave
  else
    Draw  $M_0 \sim \text{Unif}(\mathcal{A}_1), M_1 \sim \text{Unif}(\mathcal{A}_1)$ 
    Draw  $U_2 \sim \text{Unif}(0, \Delta)$ 
    if  $(U_2 < d(M_0, M_1))$   $\phi = move$ ; leave
  end do

```

**Figure 2. Sampling algorithm for phase  $\phi$ , previous and next trip endpoints  $M_0$  and  $M_1$ .  $\tau_{pause} := \int s f_S^0(s) ds$  is the average pause time and  $\omega = \int \frac{1}{v} f_V^0(v) dv$  is the mean of the inverse of the numerical speed, sampled on an arbitrary trip. For space graph  $\mathcal{A}_1$  is the set of vertices, and  $\Delta$  is average distance between vertices of the graph. For city-section  $\mathcal{A}_1 = \mathcal{A}$ , and  $\Delta$  is an upper bound on diameter of  $\mathcal{A}$ .**

The perfect sampling algorithm in Figures 1 and 2 illustrates a merit of the generic perfect sampling proposed in [6]. The benefit is that the algorithm does not require knowing geometric constants, such as, in the city-section example, the average distance between two random points on a graph, which may be difficult to compute. It suffices to know an upper bound on the distance between any two points in the domain, which for the city-section is a non problem to compute.

### 3 Implementation of Perfect Sampling

In this section, we describe random trip mobility models for which we implemented perfect sampling. The models are: random waypoint, random walk with wrapping, random walk with reflection, restricted random waypoint and space graph as a particular instance of restricted random waypoint. We briefly describe each of the models. For more complete definitions, we refer to [6].

Our tool generates a mobility trace file in ns-compatible format, and thus can be directly used as input to ns-2 network simulator. For concreteness, the format consists of:

```
$ns_ at TIME "$node_(NODE) set X_ X1"  
$ns_ at TIME "$node_(NODE) set Y_ Y1"  
$ns_ at TIME "$node_(NODE) set Z_ Z1"  
$ns_ at TIME "$node_(NODE) setdest X2 Y2 SPEED"
```

which set the position of a node `NODE` to  $(X_1, Y_1, Z_1)$  at time `TIME`. The trip destination point of the node `NODE` is  $(X_2, Y_2, 0)$  and numeric speed is `SPEED`. The above command primitives are sufficient to define various mobility patterns. We next define random trip models, for which our tool produces perfect sample:

**Random waypoint:** At a trip transition instant, a node picks a trip destination uniformly at random on a rectangular area and samples numeric speed from a uniform distribution. The trip path is the straight line that connects node positions at this and next trip transition instant. Upon reaching the trip destination, the node may pause for a random time drawn from a uniform distribution. This trip selection rule repeats. A default initialization rule is to set the node at time 0 to either move or pause phase and specify time 0 as a trip transition instant. This model has been extensively studied before in literature, and is the basis for most of the evaluations of ad hoc routing protocols. A steady state version of random waypoint has been implemented by Navid and Camp [5]. Our implementation uses a different perfect sampling algorithm [6], which alleviates knowing geometric constants. The perfect sampling algorithm is that of Figures 1 and 2.

**Random walk with wrapping:** This model is similar to the random waypoint, but at a trip transition instant, a node picks direction, trip duration, and numeric speed. The node moves in the given direction with the given numeric speed for the given trip duration. If on a trip, the node hits the boundary of the domain, it is wrapped around into the domain. The steady state version of this model is such that node position is uniformly distributed on the domain, the node speed has the same distribution as at a trip transition instant. Conditional on phase is *move*, the node position, speed, and residual trip time are independent. Conditional on phase is *pause*, the node position and residual pause time are independent. See [6] for more rigorous statements. The implemented perfect sampling algorithm is that of Theorem 10 in [6].

**Random walk with reflection:** The difference between this model and random walk with wrapping is that, whenever a node hits the boundary of the domain, it is not wrapped around, but reflected into the domain.

**Restricted random waypoint:** The domain contains a specified collection of sub-domains. At some trip transition instants, a node performs Markov walk on the sub-domains, for other trip transition instants the node undergoes trips within a sub-domain. The number of trips within a sub-domain is drawn at random at a trip transition instant for which the node enters a sub-domain. When the number of sub-domains is equal to 1, the perfect sampling is given by Figures 1 and 2; for the general case it is that found in [6].

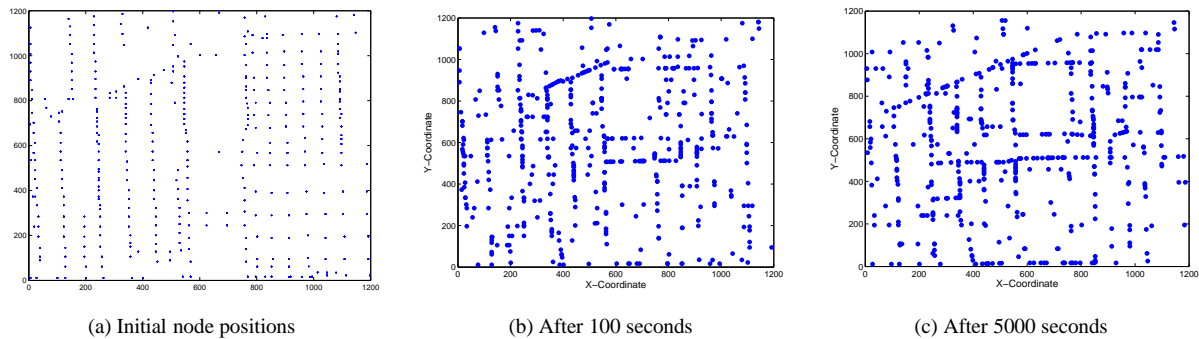
**Space graph:** This is a particular instance of restricted random waypoint. We are given a space graph, which is a collection of graph vertices, each associated with a point in space, and a connectivity matrix that specifies edges between the vertices. The domain is the union of line segments defined by the graph edges. The sub-domain is the set of the graph vertices. Path between two vertices is a shortest path.

**City section:** This is a particular instance of random waypoint on a general connected area. The domain is the union of line segments defined by the edges of a given space graph. The perfect sampling algorithm is that of Figures 1 and 2.

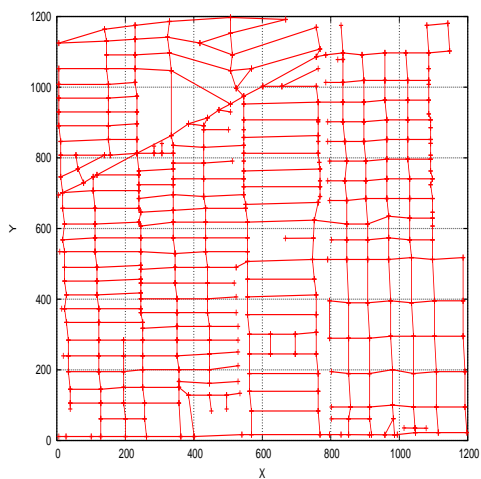
We implemented perfect sampling for the above random trip models in the tool `randomtrip`. The tool takes input data that specify a particular random trip model instance, and thus provide a user with flexibility in defining a particular mobility scenario. The common inputs are the number of mobiles and simulation duration. Other model-specific input parameters are described here:

- For random waypoint, random walk with wrapping and reflection, the input data include side lengths of a rectangular domain, parameters that specify distribution of the numeric speed for the random waypoint; for random walk models the input data are parameters that specify trip duration.
- For space graph and city-section models, the input is the space graph defined in an ASCII format that contains: road id, average road speed, coordinates of the road endpoints.
- For restricted random waypoint, the input data include the number of sub-domains and specification of the sub-domain geometries from a set of elementary shapes in an ASCII format.

The output of each tool is ns-compatible ASCII file that specifies node movements for a simulation.



**Figure 4. Distribution of nodes in the graph with random uniform pause between 0 and 100 seconds.**



**Figure 3. Realistic street scenario corresponding to a square area of size 1200 m $\times$ 1200 m. The scenario consists of 383 intersections and 594 road segments.**

## 4 Simulation Results

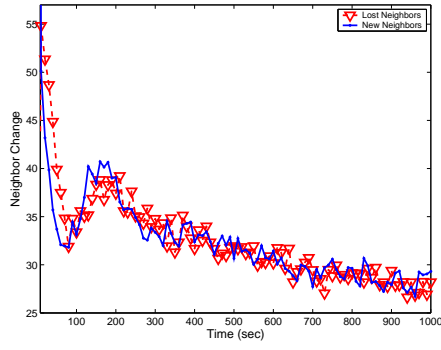
In this section, we evaluate by simulations some mobility properties for a space graph mobility model. In particular, we consider some performance metrics that characterize node communications, such as for example those that effect ad hoc routing performance. Given a realization of node movements, we evaluate node communication properties such as the size of the node neighborhood set (node degree), average and maximum path distance between node pairs, fraction of disconnected node pairs and their variation with time. These performance metrics would correspond to performance of a hypothetical routing algorithm that posses global knowledge of the network state. We assume a fixed wireless transmission range model; this is a simple standard wireless communication model specified by a single param-

eter  $r$ , the wireless transmission range. With this communication model, any two nodes can communicate if and only if the distance between the two nodes is at most  $r$ . We deliberately evaluate the performance metrics for the hypothetical routing protocol, rather than for an ad-hoc routing protocol, in order to isolate fundamental mobility-originated performance from those due to various artifacts associated with a particular ad hoc routing protocol.

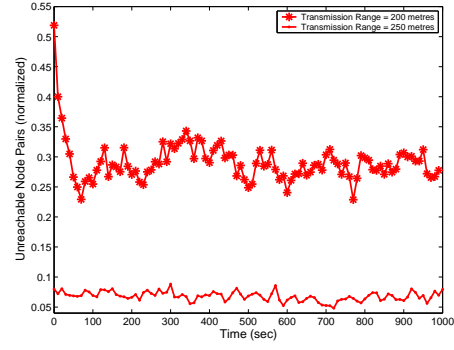
However, in addition to simulation results for the hypothetical routing algorithm, we do show simulation results for DSR [10], a widely-known ad-hoc routing protocol. The aim of this simulation set would be to verify the claim: In many cases, perfect simulation of mobility induces a smaller duration of the transient phase for the whole system, than if the mobility simulation is non perfect. Indeed, a perfect simulation of node mobility does not necessarily imply perfect simulation of the whole system. If the claim is true, then it is a strong incentive to run perfect mobility simulations. A complete verification of the above hypothesis would be to run simulations for an extensive set of scenarios. We do not provide a complete study, but a limited set of simulation evaluations that do not contradict the claim (showed later). A complete hypothesis test is left open for future work.

We instantiate a sample of a space graph as follows. We used a real-world road map of a section Houston close to Rice University.<sup>1</sup> The area of the section is roughly a square with a side of length 1200 meters. The area of concern may be regarded residential with perpendicular street intersections and a low speed limit shown in Figure 3. We simulated movements of 50 mobiles. The numeric speeds were chosen with uniform distribution on the interval from 0.01 to 9.99 meters per second. This was approximately for 5 meters per

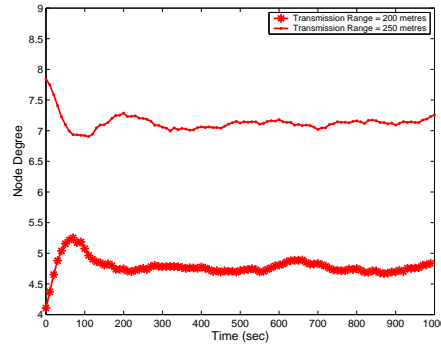
<sup>1</sup>The detailed maps are available from the United States Census Bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) database. These files are freely available to the public for use, and are typically used to provide the digital map base for all Geographical Information Systems and route finding software. This tool was popularized recently in [11], and we leverage this database to provide us with realistic city maps.



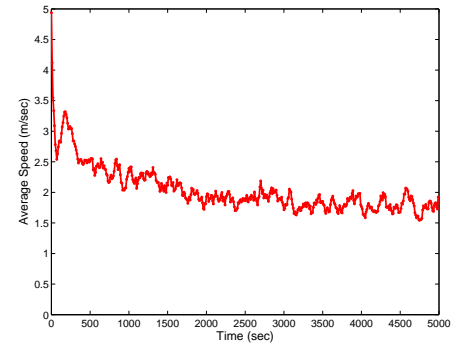
(a) Neighbors change



(b) Unreachable nodes (normalized)



(c) Average node degree



(d) Average node numeric speed

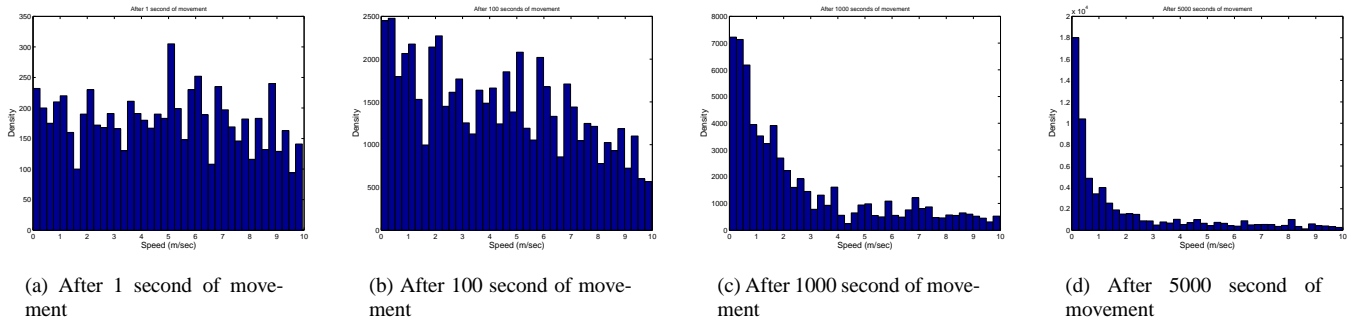
**Figure 5. Graph characteristics for movement with uniform pause from 0 to 100 seconds.**

second above and below the fixed speed limit of 5 m/sec for each road. We chose the lower speed to be strictly positive in order to have a finite average trip duration as seen at a trip transition instant. We ran all our experiments for 10 runs, each of duration 5000 seconds and each with the initial mobility state drawn by using a different seed for the random number generator. We averaged a value of concern over all the runs. We also ran all the simulations with 100 nodes, but omit those results as they are essentially similar to the 50 node simulations. Next, we present our simulation results with pause (results with no pause are omitted due to lack of space), and then present the simulation results obtained with DSR routing protocol.

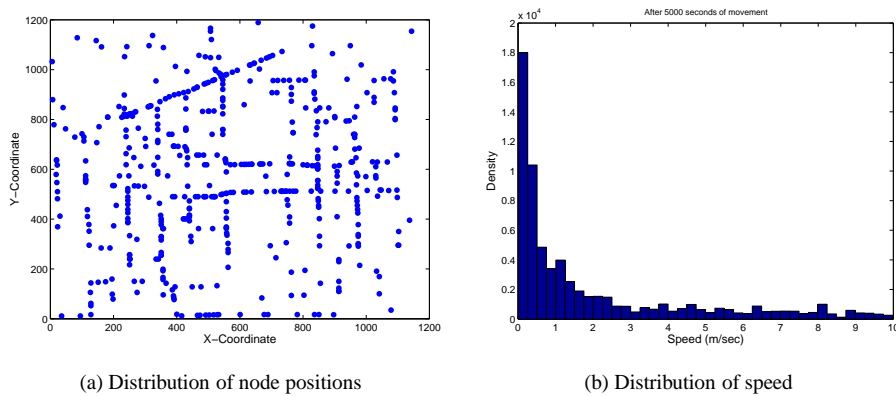
For each mobile the pause time is random, uniformly distributed between 0 and 100 seconds. We assume the default initialization rule that specifies each mobile initially to be in move state, and then alternates between move and pause. We first consider mobile positions in Figure 4 at initial state and after 100 and 5000 seconds since the simulation start. We note that the graph may be cut into two subgraphs by a cut at about  $x = 700$  meters. In the region around  $x = 700$  all the roads are dead-end, except 4 roads. In Figure 4(c), we note that the region around  $x = 700$  is sparsely populated by mobiles in the initial state, while in steady-state it

is more densely populated.

We now consider variation of node neighborhood set with time. This metric may have effect on ad-hoc routing protocol overhead. We consider the increments and decrements of the size of the node neighborhood set, defined as follows. For each 10-second interval, we calculate the number of the neighbors that were reachable in the previous 10-second interval and are no longer reachable in the current 10-second interval. Similarly, we count the number of new neighbors a node encounters on a 10-second interval, which were not reachable by the node in the previous 10-second interval. Denote the former and latter increments, as  $\xi_n^{old}$  and  $\xi_n^{new}$  for the  $n$ -th 10-second interval, respectively. Then, the number of the neighbors  $N_n$ , in an  $n$ -th 10-second interval, evolves as:  $N_{n+1} = N_n + \xi_n^{new} - \xi_n^{old}$ . The empirical averages in Figure 5(a) show that the two neighborhood increment elements have virtually the same means, for a given time instant. The neighborhood increments decrease with time and eventually settle around a fixed point, which corresponds to the steady-state average. The duration of this convergence time is significant. A plausible intuition is that the node neighborhood set change would be highly correlated to the value of the node numeric speed at a time instant. This is confirmed later in Figure 6.



**Figure 6. Distribution of speed. Uniform pause - 0 to 100 seconds**



**Figure 7. Perfect Simulation (steady state initialization) at time 0. Uniform Pause - 0 to 100 seconds**

We next consider the fraction of disconnected node pairs. The results are shown in Figure 5(b). We show the results for wireless transmission range set to 200 and 250 meters. We lower the wireless transmission range to 200 purposely being led by the sparsely populated region around  $x = 700$  (noted earlier) for the initial node positions. The results demonstrate that connectivity properties may be dramatically different in transient phase and in steady-state (see the sharp drop of the fraction of disconnected nodes at the origin in Figure 5(b), for the wireless transmission range 200 meters).

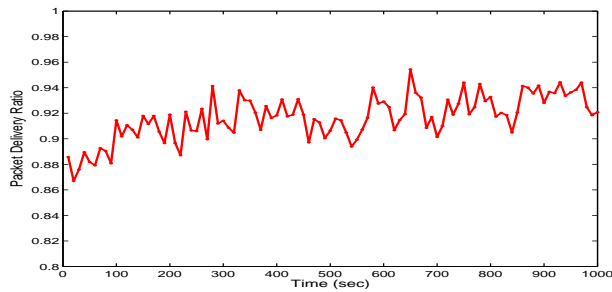
We further show the average node degree in Figure 5(c). We suspect that virtually constant value of the average node degree is due to large wireless transmission range, relatively with respect to the area of the simulation and density of nodes. The smaller average node degree for the initial node positions is due to smaller mass in the distribution of node degree to small degrees.

We consider steady-state average numeric speed as computed at the instants of a simulation as in Figure 5(d). The average numeric speed decreases with time, as predicted by the analysis, and takes more than 2000 seconds to reach a neighborhood of the steady-state average speed. The average speed at the start of the simulation is 5 meters per

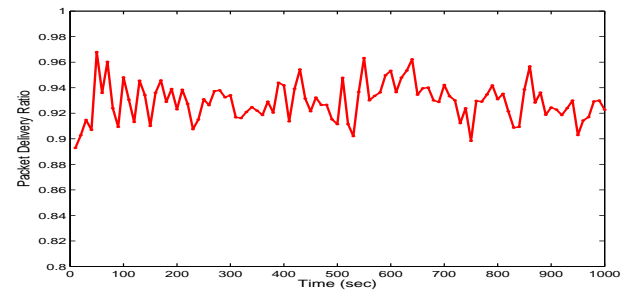
second, which is the average of a random speed with uniform distribution on the interval from 0.01 to 9.99 m/sec. The steady-state average speed is somewhat smaller than 2 meters per second. We complement the average numeric speed in Figure 5(d) with empirical histograms of the numeric speed at the origin of a simulation, after 100, 1000, and 5000 seconds from a simulation start, see Figure 6. Initially, all the node numeric speeds are uniformly distributed on the interval from 0.01 to 9.99 meters per second. The larger the time in a simulation, the closer the distribution of the numeric speed to the steady-state distribution of the numeric speed. The results indicate that this convergence time is larger than 1000 seconds, and it seems to virtually converged at 5000 seconds from a simulation start (Figure 6(d)).

We now show the node positions and empirical distribution of node numeric speeds obtained by our perfect sampling in Figure 7. Both visually conform well with the respective results showed in Figures 4(c) and 6(d), obtained for large time of 5000 seconds. Thus, indicating perfect simulation and hence validity of our perfect sampling.

Next, we consider performance of DSR routing protocol on these scenarios. We configure 20 data connections, each with the constant bit rate of 1 packet per second and packet



(a) Non perfect mobility simulation



(b) Perfect mobility simulation

**Figure 8. Packet delivery ratio achieved by DSR protocol when the mobility simulation is non perfect and perfect.**

length fixed to 512 bytes. Each connection is initiated at a time instant, which is drawn independently, uniformly at random on the interval 0 to 40 seconds of the simulation time. Each connection lasts until the simulation ends. We consider a performance metric defined as follows. Over 10-second time intervals, we compute the ratio of the total number of packets received by the nodes and the total number of packets sent. We call this ratio packet delivery ratio. The results are showed in Figure 8 for both non perfect and perfect simulation of the node movements. With non perfect mobility simulation, there is a notable initial transient for packet delivery ratio; when the mobility simulation is perfect, this is not noticable.

## 5 Conclusion

We described our implementation of perfect sampling for a set of random trip mobility models. We also present simulation results for a particular random trip mobility model that we construct from existing road maps, which thus captures some aspects of reality. We evaluate several node communications properties and their dependence on the underlying mobility. The main observations from our simulation results are: (i) transient phase may last long, longer than a typical simulation duration, (ii) the transient phase effects performance, which is different in steady-state, (iii) our perfect sampling produces perfect simulations of node mobility. Our tool for perfect sampling produces input to widely-used ns-2 network simulator, and is available to public.

## References

- [1] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta G. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*. ACM/IEEE, Oct. 1998, pp. 85–97.
- [2] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. 2003, pp. 217–229, ACM Press.
- [3] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communication and Mobile Computing (WCNC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.
- [4] Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful," in *Proceedings of Infocom*, 2003, vol. 22, pp. 243–254.
- [5] William Navidi and Tracy Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, Jan-Mar 2004.
- [6] Jean-Yves Le Boudec and Milan Vojnović, "Perfect simulation and stationarity of a class of mobility models," in *Proceedings of IEEE Infocom 2005 (to appear)* - also available as *EPFL Technical Report IC/2004/59*, March 2005.
- [7] Jorjeta G. Jetcheva, Yih-Chun Hu, Santashil PalChaudhuri, Amit Kumar Saha, and David B. Johnson, "Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture," in *Fifth IEEE Workshop on Mobile Computing Systems and Applications*. Oct 2003, pp. 32–43, IEEE Press.
- [8] "Census bureau home page," Available from <http://www.census.gov/>.
- [9] Jean-Yves Le Boudec, "Understand the simulation of mobility models with palm calculus," Tech. Rep. EPFL/IC/2004/53, EPFL, 06 2004.
- [10] David B. Johnson and David A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Tomasz Imielinski and Hank Korth, Eds., chapter 5, pp. 153–181. Kluwer Academic Publishers, 1996.
- [11] David B. Johnson Amit Kumar Saha, "Modeling mobility for vehicular ad-hoc networks," in *Proceedings of the first ACM workshop on Vehicular ad hoc networks*. Sep 2004, pp. 91–92, ACM Press.