A Framework for Evaluating DTN Mobility Models

Agoston Petz, Justin Enderle, and Christine Julien
Mobile and Pervasive Computing Group
University of Texas at Austin
{agoston, justin.enderle, c.julien}@mail.utexas.edu

ABSTRACT

The field of delay tolerant networking is rich with protocols that exploit node mobility to overcome unpredictable or otherwise bad connectivity. The performance of many of these protocols is highly sensitive to the underlying mobility model which determines the nodes' movements, and the characteristics of these mobility models are not often studied or compared. With few exceptions, authors test their ideas using mobility models implemented on simulators written for the specific purpose of testing their protocols. We argue that it is better to unify these simulations to one highly capable simulator. We develop a suite of mobility models in OMNeT++ that specifically target delay tolerant networks. We also present a series of metrics that can be used to reason about mobility models independent of which communication protocols and data traffic patterns are in use. These metrics can be used to compare existing mobility models with future ones and also to provide insight into which characteristics of the mobility models affect which aspects of protocol performance. We implement a tool that derives these metrics from OMNeT++ simulations and implement several popular delay tolerant mobility models. Finally, we present the results of our analysis.

Categories and Subject Descriptors

I.6.7 [**Simulation Support Systems**]; C.2.1 [**Network Architecture and Design**]: Store and forward networks

Keywords

Mobility Models, Delay Tolerant Networks, OMNeT++

1. INTRODUCTION

Delay tolerant networks (DTNs) are steadily gaining popularity in the research community for their ability to provide connectivity, or a semblance of connectivity, in "challenged" networking environments. Examples of these environments include (1) urban networks in which opportunistic meetings between cars and buses can be used to transfer messages from disconnected portions of the network to areas with Internet access [8]; (2) rural networks in which villages have reliable connectivity between local hosts but have unreliable connections to the wider world [22]; (3) networks of sensors that collect and share information about animal movement and behavior [15]; and (4) networks in which roving autonomous robots provide connectivity or message ferrying capabilities in disruption-tolerant environments [21, 3]. In all of these networks, connections appear and disappear on an unpredictable schedule, are sometimes only available for a very short amount of time, and the source and destination of any given end-to-end communication may never be directly connected.

These different types of environments have their own set of challenges, and solutions have been developed for many of them. However, often these solutions are sensitive to the movement patterns of the underlying nodes and rely on a particular vision of how the mobile nodes behave. Many papers make use of the random waypoint mobility model, which is not the best interpretation of node movement since real nodes rarely behave in a completely random fashion [11, 30]. Random waypoint does have benefits: it is mathematically analyzable and easy to implement. Thus, it is easy for researchers to test their own algorithms using it, which makes results between papers comparable. In many other cases, the mobility models used in one paper are never seen again, and it is difficult or impossible to reason about how the characteristics of the mobility model would affect the performance of a different routing algorithm or protocol.

Most research in the area of delay tolerant networking relies on simulation to validate ideas since real-world deployments are often either very expensive or impossible. The results of these simulations are obviously sensitive to the level of realism in the simulators, many of which do not implement a realistic radio model or networking stack.

There is also an issue of cross-paper comparability—researchers often create their own simulators to test their algorithms, and it can be very difficult to compare a new algorithm with existing ones unless you implement it on a variety of simulators. In the same way, it is difficult to make use of existing mobility models. In this paper, we argue that it is important to have a standard suite of DTN mobility models that can be made available to researchers. Since we want to make it easy for others to use our mobility models and to contribute new ones, we chose to implement them in OMNeT++ [27], a widely available and capable open-source simulator. To justify this decision, we later demon-

SCENES 2009 Rome, Italy

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

strate that the performance of OMNeT++ for large-scale simulations is within acceptable bounds. We provide OMNeT++ implementations of several popular mobility models (Section 3). Additionally we create a statistical analysis package for OMNeT++ that greatly extends the capabilities of the simulator and provides insight into the characteristics of the mobility models independent of the overlying communication protocols (Section 4). These statistics can provide insight into how different mobility models affect ad-hoc or DTN routing algorithms and other higher layer protocols. Due to the modular nature of OMNeT++, authors of new mobility models do not have to make any changes to their code to use our statistics package and benefit from the information it provides. Finally, we provide the metrics by which different mobility models can be compared, and we present our analysis of the mobility models we implemented (Section 5). Section 6 gives future work and concludes.

2. BACKGROUND

Related work can be divided into two categories: work on mobility models seeking to increase their realism, and work on simulators aimed at gathering information on existing scenarios to provide insight into the characteristics of mobility models and how they affect the performance of routing protocols.

2.1 Modeling Mobility

Communication in a Delay Tolerant Network is often facilitated by the mobility of the devices, as it is through their movement that initially unreachable hosts become connected. Messages are delivered to initially unreachable devices by waiting for a direct contact opportunity or by forwarding the messages between reachable devices with the aid of DTN routing algorithms. A number of the routing algorithms proposed are agnostic towards the device mobility patterns [10, 26, 29], however as the DTN community recognizes the impact of mobility on performance, a number of algorithms have been proposed that attempt to enhance delivery and network efficiency by using the non-random mobility patterns in the environment to make better routing decisions [17, 18].

With mobility dependent protocols being developed, a wide variety of mobility models have been proposed to try and capture more realistic movement patterns based on anticipated scenarios. Many of the models attempt to capture the social patterns of human mobility, while others explore transportation systems and animal mobility. Lindgren et al. [17] propose a community modeled as a set of grid positions that nodes switch between, with a home grid position assigned to each device and a central gathering grid common to all devices; nodes move primarily between home and the central gathering place. The Working Day Movement Model [7] is a more sophisticated model with many of the same ideas, while also adding a time scale to switch between sub-mobility models and locations for home, work, and evening activities. A variety of transportation methods are available between the locations as well. The Community Based Mobility Model [19] represents the social dynamics between hosts as a graph with weighted edges indicating the strength of interaction between hosts and determines movement between grid positions based on the affinity to interact with the hosts currently located there. The authors from [28] use both the Fixed Route Campus Bus Model and Area-Based Random Waypoint Model for evaluation. The bus model simulates a network of buses traveling between fixed points on a college campus, with data transferred between buses when they are stopped at the same location. The Area-Based Random Waypoint model assigns a specific sub-area to each host within which the nodes move according to standard random waypoint mobility model with no restrictions made on how the host's areas overlap. The Obstacle Mobility Model [11] models the obstructions that exist in the environment, and determines the paths used to move around them. The mobility patterns of zebras have been studied as well [15].

While the previous models proposed were created in an attempt to model reality, an alternate approach is to collect real world mobility statistics for a network and build models that fit the patterns observed from the collected data. Traces have been taken at a variety of locations and then used either as direct inputs for mobility to a simulation, or used to generate a synthetic mobility model matching the characteristics of the trace data [5, 31, 24]. Developing models from contact traces can be difficult, as the precision of the position and contact information collected limit the granularity in which mobility can be detected, limiting the ability to capture fine-grained mobility characteristics.

2.2 Simulating Mobility

The seminal work on comparing different mobility models with the intent of providing insight into how different models affect routing performance compared several random models with Reference Point Group Mobility, a Gauss-Markov model, a city section mobility model, and many others) [4]. However, the statistical analysis methods have not been applied to mobility models developed more recently, several of which exhibit greater degrees of realism. Baumann, et al. developed a Generic Mobility Simulation Framework (GMSF) [2] for the purpose of comparing different mobility models aimed at vehicular communication networks. GMSF is capable of exporting simulation traces to a variety of different formats such as ns-2 and Qualnet and collects a number of statistics that can be used to reason about the mobility models. GMSF can also handle a number of different radio propagation models. However, GMSF omits a number of statistics that are important to delay tolerant networks such as network partition sizes and memberships. We also note that researchers who desire to make use of GMSF's analysis, but desire a more realistic radio model or a more capable simulator (such as ns-2, or Omnet) are then forced to run their mobility models first through GMSF, then export traces to their choice of second-stage simulator. This is not necessarily the best method since it introduces an additional dependency, and small changes to the mobility model become more cumbersome to enact. It is also uncertain how small differences in each simulator's interpretation of node movement and radio connectivity affect the results. We argue that it is better to collect statistics for both mobility and communication in a single simulator simultaneously.

Our desire to collect comprehensive statistics to characterize the differences between mobility models is by no means unique. BonnMotion [6] implements several popular mobility models including Random Waypoint, Gauss-Markov, the Manhattan Grid Model, and the Reference Point Group Mobility model and provides a comprehensive statistical analysis of each. However it is not a simulator, but rather a

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

trace generator and mobility model analysis tool. It would be difficult to implement any kind of routing algorithm on it, and it has a very naïve uniform disk radio model.

Keränen and Ott's ONE simulator is another capable and popular DTN simulator [16]. It allows for city maps to be used to generate realistic city movement models and implements several popular delay tolerant routing algorithms. However, it does not include a realistic radio model, nor does it include any kind of network stack. DTNSim [10] and DTNSim2 [14] suffer from the same problems.

2.3 Motivation

Given the existing work on mobility models, we answer two fundamental questions that drive the remainder of the work reported in this paper.

Why a networking stack? A realistic network stack is important for several reasons. First, it allows researchers to simulate real applications which expect certain transport and network protocols to be present in order to function, and thus it allows the simulator to model the interaction between higher layer protocols and the DTN. The ability to directly simulate higher layer protocols prevents researchers from making invalid assumptions about protocols' potential behavior. Second, there is evidence that real DTN deployments will handle a mixture of TCP/IP protocols and DTN protocols and that routing in such networks will consist of a hybrid of existing ad hoc networking protocols and new DTN protocols [20]. Allowing both to coexist in the simulator is the first step to understanding their interactions.

Why use OMNeT++? In our initial search for a capable simulator with native mobility support, we came across several possibilities: GloMoSim [1], ns-2 [13], ONE [16], and DTNSim2 [14]. While simulators like ONE are very fast, easy to use, and provide great support for defining mobility models based on real-world scenarios, they lack much of the functionality available in more capable simulators like GloMoSim, ns-2, or OMNeT++. While OMNeT++ by itself is not as versatile as ns-2 or GloMoSim, the inclusion of the INET framework (an optional, but fully supported add-on) makes it one of the most capable simulators. OMNeT++ with the INET framework has a full TCP/IP network stack, and our initial experience with it is very positive—it is highly modular, very powerful, and easy to use. The architecture of OMNeT++ allows users to change radio models, routing protocols, applications, and virtually any component of the simulation without recompiling. It is written in C++, and thus much of the code needed to implement new protocols in OMNeT++ is re-usable for implementing the same protocols on real nodes.

3. SAMPLE DTN MOBILITY MODELS

In addition to the random waypoint mobility model, we implemented three DTN-specific mobility models to evaluate the mobility statistics package we developed. It is not our goal in this work to evaluate the accuracy of the models as implemented, but instead to ensure that they capture the spirit of the intended mobility patterns, and evaluate how the statistics gathered from our framework can allow these models to be compared using metrics meaningful to DTN performance.

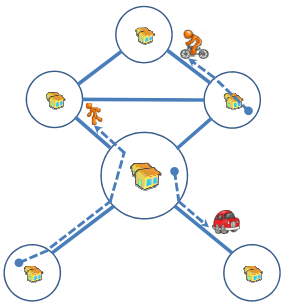


Figure 1: Village Mobility example

3.1 Zebra Mobility

Our implementation of zebra mobility is based on the ZebraNet paper [15], which gives detailed information regarding the specific mobility habits of zebras. Each zebra in the model moves independently in a landscape composed of rectangular grazing areas and watering holes. A zebra's general movement pattern is a random waypoint search for a grazing area, interspersed with periodic trips to a watering hole. In searching for a grazing area, zebras use the *roaming mode*, in which they move faster and across longer distances following the random waypoint model. Once a zebra finds a grazing area, it enters *grazing mode*, moving slower and across shorter distances between each movement. The speed for grazing and roaming is set by a base grazing speed, with a constant speedup factor for roaming. After each movement in a grazing area, the zebra randomly decides to continue grazing or return to the roaming mode, selecting a random position among the landscape as the next destination. Each zebra has a hunger parameter, which determines the likelihood for the zebra to continue grazing.

The mobility within grazing areas is modeled after observed patterns [15]. After each movement, the distance of the next movement and a change of heading angle is selected. The distribution of the distances between movements was shown to be bimodal, with the two peaks having means of 3.1 and 13 meters. This is approximated by drawing values with equal likelihood from normal distributions using the same means and a variance of 1.0. Intuitively, this means grazing movement results in either short or mid-range movements with equal likelihood. The angular change from one movement to the next is similarly drawn from a normal distribution centered at 60° with a variance of 1.0. All distributions are truncated to remove values less than zero, with the angular values truncated above 180°.

Each zebra also regularly visits a watering hole by maintaining a timer that causes the zebra to move directly to a watering hole at roaming speed. After reaching the water, the zebra resumes roaming mode in search of a grazing area.

3.2 Village Mobility

A village mobility scenario consists of the villages and people who inhabit them. The villages are dispersed in the landscape, with villages below a threshold distance from each other directly connected by roads. These roads create a transportation network by which people can move between villages. Figure 1 shows an example network of villages and the movement of people between them. The circle surrounding a village represents its area; people within the village move in this circle, and the areas of nearby villages

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

can overlap.

Each person (host) is initially assigned and randomly placed within a home village that it remembers throughout the simulation. The number of hosts assigned to a village can be used to set its radius size, creating larger areas for higher populations. Village populations also influence the choice of village destinations, described next.

A host chooses between three options when picking its next destination. The host chooses to move to its home village, stay at the current village by picking a random position within its area, or randomly visit a non-home village. If a host chooses its home village, and it is already there, it picks a position within its radius as well. Parameters are used to modify the likelihood of the different options.

When moving to a different village, any village that has a connected path through the road network is a valid choice, with Dijkstra's shortest path algorithm used to determine the path of roads and villages to traverse to reach the final destination. The choice of the next village can be completely random, or instead weighted proportionally to its population size. A move to a new village comprises a number of separate movements. When moving from village to village, the host must first move to the road entrance at the edge of the village which connects the villages together. From there, it moves straight along the road to the next village, and repeats the process if additional hops through villages are necessary.

All movement within a village is done by walking at a set speed, but hosts travelling between villages can also use bicycles and cars traveling at higher speeds. The likelihood of choosing any given method of transportation is proportional to the distance between the starting point and the destination. For example, the probability of choosing a car, $P(car)$, is the ratio between the distance to the next village and the maximum possible distance between villages. The probability of walking or riding a bicyle is thus $(1 - P(car))$ and $P(bicycle)$ is always twice $P(walk)$, thus there is a 2:1 ratio between biking and walking. To ensure a non-negligible probability of each travel type, $P(car)$ has a max of 0.8 and min of 0.2.

3.3 Truncated Levy Walk Mobility

The Truncated Levy Walk model is a purely statistical model which draws values from a random distribution to determine the distance traveled and angle of movement for each new destination, as well as the pause time between movements. It uses a power law distribution $p(l) \sim \frac{1}{l^{1+\alpha}}$ [24] with $0 < \alpha < 2$ to generate the flight lengths, which is a heavy-tailed distribution causing a majority of the flight lengths to be short, but with occasional long flights occurring as well. Angles of movement are pulled from a uniform distribution. Truncated Levy Walk is a type of power law distribution that has been studied extensively for animal patterns and recently has been shown to be promising as a model for human mobility [9, 24]. Our implementation is directly based on the model by Rhee et al. [24].

The primary characteristics of the mobility can be changed by varying the power law exponent, α, and the truncation factor, τ. The α parameter changes the ratio of short flights to long flights: lower values of α cause longer flights. Modifying the distribution of flight lengths greatly impacts the mobility pattern, as an increased number of long flights allows nodes to travel further in the same period of time. Figure 2 shows the cumulative distribution function of the displacement between the start and end points of a simulation, showing much higher displacement for lower α values. The truncation factor, τ prevents flight lengths above a threshold value. Intuitively, this represents the real-world limitations imposed by our environment, such as obstacles limiting flight length and the inability and unlikelihood of remaining still for extremely long periods of time.

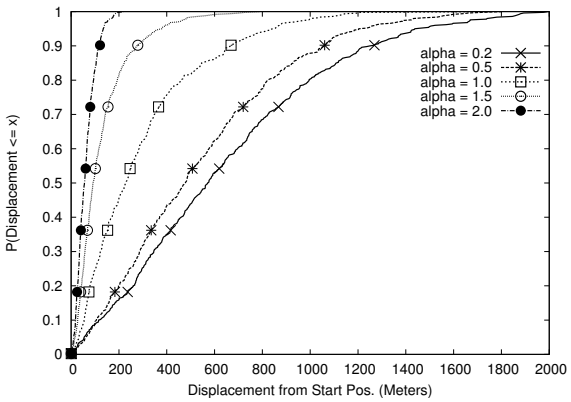


Figure 2: Displacement CDF on a 2km x 2km grid over 10 minutes with flight truncation = 1km, pause truncation = 1000s, and pause α = .5

4. STATISTICS MODULE FOR OMNET++

To gain a better understanding of the fundamental characteristics of mobility models for delay tolerant networks, we developed a statistics package to collect data about relevant aspects of mobility. This package not only allows us to evaluate the aforementioned models, but it enables us to compare different models on axes relevant to the success of DTN communication protocols.

4.1 Design

We designed the statistics module to help us reason about how characteristics of a given mobility model affect the performance of routing and other higher layer protocols. We also intend for these statistics to enable us to draw conclusions about how mobility models are similar and how they differ. For example, it would be interesting to discover that two models that appear to be quite different on the surface result in similar values for those metrics which end up governing the performance of a given routing protocol.

Our first iteration focuses strictly on mobility characteristics and does not collect information from the specific radio model (for example by querying the radio for SNR or dropped frame information) or any networking protocols or applications which may be in use on a node. This is a benefit in that it makes the module highly portable, and no code changes are necessary to use it. The only changes which are required are changes to the network definition files and a global configuration file. However, this is also a drawback in that the statistics module cannot track buffer space, signal-to-noise ratios, or a variety of other possibly interesting variables. Since our statistics module is independent of any specific radio implementation, it cannot reason about connectivity—instead we consider *possibilities* for connectivity, and *opportunities* for message exchange given an ideal

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

radio range. Since we do not implement any specific physical radio model, we allow users to define a contact "range," and if two nodes are in range of each other, we record their potential interaction. We also assume that any given contact can result in a successful message exchange. This is obviously not always true, but it is useful data in generating an upper bound on communication potential and helps us better understand the characteristics of different mobility models. In our next version we hope to query the actual physical radio module to provide connectivity information in addition to the ideal contact data we currently provide.

The statistics module currently collects the following statistics for every node in a network:

Core Statistics: The statistics module samples each node on a user-defined interval. It collects the following basic statistics for each node in the network and records their time-varying values and their averages: node position, number of neighbors, and number of unique neighbors (which indicates how many other unique nodes the node encountered during the course of the simulation).

Connection and Partition Tracking: Additionally, the statistics module records possibilities for connections, or "contacts", every time two nodes come within ideal radio range of each other. It records the number of these contacts over the lifetime of the simulation and their average duration per node. It also tracks the number of partitions in the network, their sizes, their memberships, and the number of disconnected nodes at any given time.

Relative Mobility: The statistics module also samples the position of all nodes on a separate user-defined timer and records the relative velocities of all nodes with respect to each other. This metric is used to calculate the total relative mobility of the network using an algorithm given in [12]. Relative mobility is a useful scalar for describing the level of degree of node movement in a network.

Message Delivery Possibilities: We have also defined a notion of potential message delivery opportunity. A given simulation can be split up into any number and duration of "epochs" during which every node starts with a unique message and attempts to deliver it to every other node. Since we are not dealing with real radios and real buffers, this is not an indication of how any specific routing algorithm would perform. Rather, we have developed an oracle that shows how a perfect routing algorithm might perform. Any contact between two nodes results in an exchange of every unique message they currently hold, and each node records the time at which it receives any message it did not already have. In this way, we establish a best case delivery latency for every message in the system and can also easily see which (if any) messages remain undelivered at the end of the simulation. Since the user can define any number of epochs for a given simulation, it is easy to see how the delivery potential changes over time due to the nature of the mobility model.

4.2 Implementation

Our statistics package is implemented as a pair of OMNeT++ modules designed to work with the mobility model architecture of the INET framework. Our modules can be dynamically linked into any simulation that includes the libraries exported by the INET framework. The first module, `MobStats`, is replicated once in each node and collects information about changes to the host's position. `MobStats` reports these changes to the second module, `StatisticsControl`, which processes the node movements into statistics. `StatisticsControl` is global to the simulation and also has a set of internal timers that determine how often it samples and calculates relative mobility, message delivery epochs, etc. Figure 3 shows the components of a simulation—we have included the `ChannelControl` module to illustrate the similarity of our architecture to that of the INET framework. The statistics module generates a set of output files that contain the time-varying data it collects and their averages. New interpretations of the data are easy to create since users are free to parse the data in any way they choose. The statistics module has a number of configuration options that determine the data it collects and how often certain data is sampled; interested parties are invited to download our code and documentation available at [23].

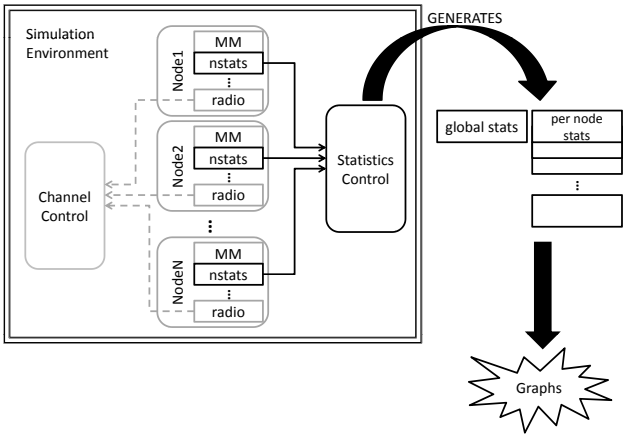


Figure 3: Architecture Overview

5. EVALUATION

We evaluated the three models we implemented and the random waypoint mobility model included with the INET framework using our statistics package. In this section, we describe the simulation setup and our results.

5.1 Simulation Setup

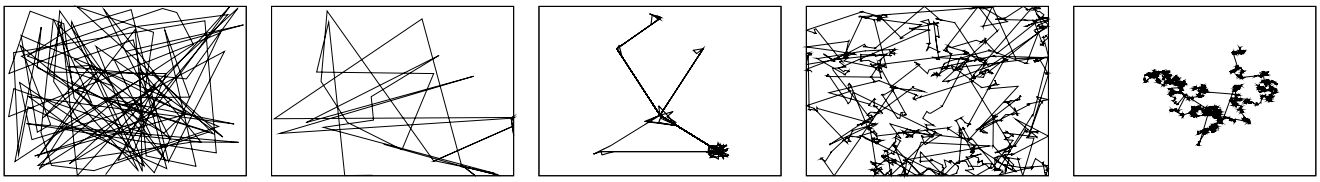
Each mobility model was evaluated with 50 nodes across a 7km x 7km landscape for a duration of 12 hours. The position of the nodes on the landscape was updated every second. A reflective boundary condition was used with all of the models to handle edge effects.

Random waypoint mobility. A constant speed of 10m/s was used for all flights. Between flights, nodes paused for 1-10 seconds (uniformly distributed).

Zebra mobility. The landscape contained two grazing areas with sizes of 1km x 3km and 3km x 3km located at [1000,1000] and [4000,4000] respectively, along with a single watering hole located at [1500,5000]. Each zebra visited the watering hole every 4-6 hours (independent and uniformly distributed with one second granularity). Grazing speed was 2m/s with a speedup factor of 5x used while roaming. Zebras paused at their destinations for 0-10 seconds (uniformly distributed) before moving towards their next target.

Village mobility. We used the villages and their relative positions as shown in Figure 1. The villages all had radii of 370m except the center village, which had a radius of 650m.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647



(a) Random waypoint (b) Zebra (c) Village (d) Levy (α=0.5) (e) Levy (α=1.5)

Figure 4: Motion traces for Random Waypoint, Zebra, Village, and Levy Walk (α=0.5 and α=1.5)

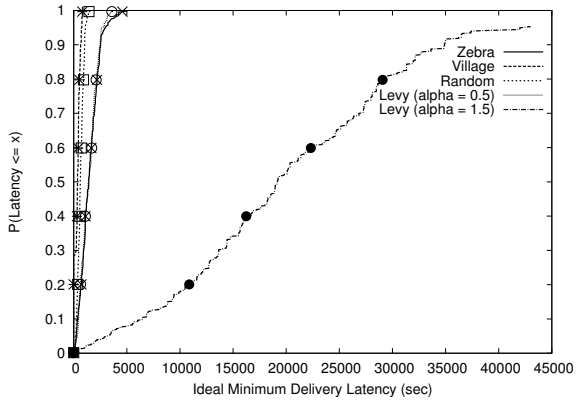


Figure 5: CDF of Ideal Minimum Delivery Latency for all simulations

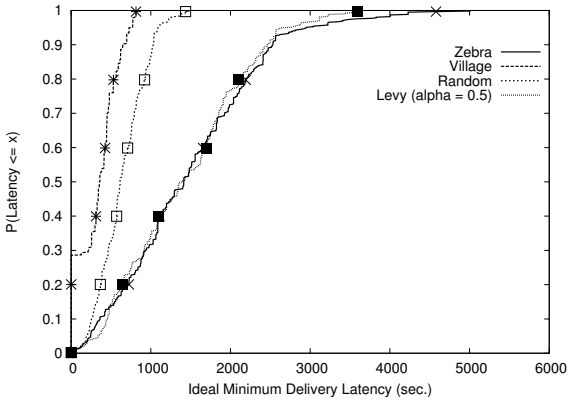


Figure 6: CDF of Ideal Minimum Delivery Latency for all simulations except Levy (α=1.5)

The maximum road distance was 3500m. The speeds used for walking, biking, and driving were 2m/s, 7m/s, and 20m/s respectively. The likelihood of each mode of transportation was set as described in Section 3.2. The center village was assigned an initial population of 25 hosts, with all of the outlying villages assigned a population of 5. This models a large central city with a periphery of surrounding small villages, with the majority of travel occurring to and from the center village from the outside ones. The probability of choosing home as a next destination was set to 0.7, staying in the current village as 0.2, and choosing a new village as 0.1.

Truncated Levy walk. We used two scenarios to show how modifications to the power law exponent α affect the mobility statistics. Although it is not required for a Levy Walk to use the same distribution for pause times, a Levy Walk distribution with $\alpha_{pause} = 1.0s$ and $\tau_{pause} = 1000s$ was used to generate pause times for both runs. The different flight length exponents α_{flight} used were 0.5 and 1.5, both with a truncation factor $\tau_{length} = 3500m$. We used a constant speed of 10m/s, and scaling factors of 10 and 1 were used for the flight and pause distributions.

5.2 Results

Not surprisingly, the mobility models resulted in very different movement patterns, as illustrated in Figure 4 which shows the path that one node took during the course of the 12 hour simulation for each of the five mobility models. Comparing Figure 4(c) and (d), one can easily see how changing the α value of the Levy Walk mobility model results in a much smaller roaming area for the nodes, and how any given node in (d) will explore a much larger area of the simulation than a node in (e). It is also interesting to note that zebra mobility results in a trace that is similar to random waypoint, but with less overall movement. The reason for this is simply that the zebra mobility model is essentially a random waypoint with a slight bias towards staying in grazing areas once a zebra encounters one.

5.2.1 Ideal Minimum Delivery Latency Performance

The ideal minimum delivery latency characterizes how fast a message *could* be delivered given the ideal message delivery possibilities calculated by the statistics module according to the metric described in Section 4.1. Any contact between two nodes results in an exchange of every unique message they currently hold. We record the time when nodes receive new messages, and this latency distribution characterizes the ideal routing performance of the network. We defined four epochs evenly spaced in the 12 hour simulation and attempted to deliver new messages at the start of the simulation, three hours in, six hours in, and nine hours into the simulation. Figures 5 and 6 show the cumulative distribution function of the ideal minimum delivery latencies of the different mobility models, averaged across the four epochs. Figure 5 includes all five mobility models to illustrate how radically different Levy walk (with α=1.5) is from the rest of the mobility models. In fact, Levy walk (α=1.5) is the only mobility model which failed to deliver every message by the end of the simulation—this is easily seen since the CDF does not reach the top of the graph. If we remove Levy (α=1.5) from the graph, we get Figure 6, which more clearly illustrates the differences between the first four mobility models. As we can see from this graph, zebra mobility

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

and Levy walk mobility (with α=0.5) performed very similarly, and were both outperformed by random waypoint and village mobility. It is interesting to note that the village mobility CDF shoots straight up to 0.3, and then levels off from there. This is due to the internally well-connected nature of the villages. At the beginning of any given epoch, co-located nodes are able to deliver their messages almost immediately, and only those nodes which are in transit or in different villages are unable to receive them until later. We did not include a graph of the CDFs of different epochs within a single simulation since none of the mobility models exhibited a significant variance in ideal minimum delivery latency performance between epochs.

5.2.2 *Aggregate Analysis*

The aggregate statistics can be seen in Table 1. From this table we can draw several conclusions. First, if we assume that a given node is equally likely to send a message to any of the other nodes in the network, Levy walk (α=1.5) presents a difficult routing problem. Since the nodes are initially uniformly distributed across the simulation space, and the roaming area of a Levy walk node is restricted to such a small area of the simulation (see Figure 4(e)), nodes are unlikely to wander far enough to meet many other nodes. Even an infinite buffer epidemic routing scheme such as the idealized "routing" we have modeled in our potential message delivery calculation would fail to deliver certain messages. Levy walk (α=1.5) nodes only had an average of 10.9 unique neighbors during the course of the simulation, compared to the greater than 40 unique neighbors posted by all other mobility models. This tends to imply an inverse relationship between delivery possibilities and the average number of unique neighbors encountered by any given node. Table 1 also implies that a low relative mobility does not play a significant role in routing, since all the other mobility models had higher relative mobility. Additionally, both zebra mobility and village mobility managed longer average potential contact durations despite their higher relative mobility metrics.

We also found that zebra mobility, which is very similar to Levy walk (α=0.5), had the second worst ideal message delivery latency distribution. Statistically, random waypoint, zebra mobility, and Levy walk are very similar, though random waypoint outperformed both in terms of ideal delivery latency. The major difference between them is the average number of contacts per hour, or the rate at which a node runs into other nodes. Since both zebra mobility and Levy walk were somewhat localized (either because of a tendency to stay in grazing areas, or due to the nature of power-law distributed flight lengths), they encountered fewer new neighbors over time, resulting in fewer opportunities to deliver fresh messages. In terms of potential routing performance, the only mobility model better than random waypoint is village mobility, which is a highly localized mobility model since nodes can only exist in certain well-defined places (within villages, or on roads). Village mobility also has the largest average partition size and the highest average contacts per hour metric (by quite a lot).

5.3 Simulation Performance

No discussion of the merits of any particular simulator would be complete without a performance analysis. In fact, performance is one of the most often cited criticisms of complex simulators such as OMNeT++; lightweight DTN simulators such as ONE will outperform OMNeT++, sometimes by several orders of magnitude. We feel that the performance hit incurred is worth the benefit of a more capable simulator. Figure 7 shows the number of simulated seconds completed per one real-time second (essentially the speed-up factor of the simulator) using our statistics module and the random waypoint mobility model for different numbers of nodes and different mobility update intervals. As can be seen from the figure, increasing the number of nodes dramatically reduces the performance of the simulator. All tests were done in a VMware environment running on a non-dedicated dual-processor 2.80GHz Pentium machine with 2 GB of RAM. Even with such a non-ideal setup, we saw very reasonable speed-up factors for simulations with fewer than 100 nodes. We also feel that computational resources are sufficiently cheap and ubiquitous that even simulations that operate at near real-time are not out of the question, depending on the desired time-scale.

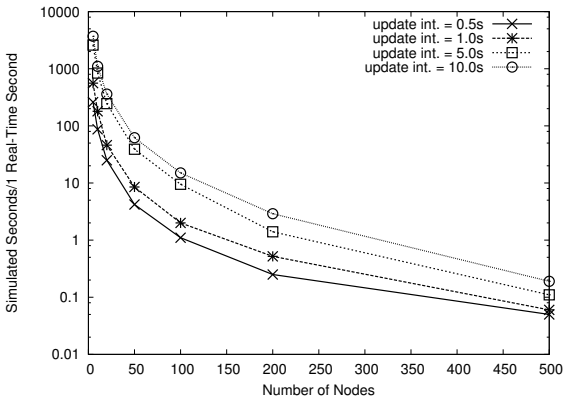


Figure 7: Performance of OMNeT++ with Statistics Module

6. CONCLUSION AND FUTURE WORK

In this paper we have argued for standardizing DTN simulations to a single capable simulator, and we have conjectured that OMNeT++ (plus the INET framework) is an acceptable option. We have implemented three mobility models—Zebra Mobility, Village Mobility, and Levy Walk Mobility—in OMNeT++ and have made our code available online [23]. We have also implemented a statistics module for OMNeT++ which, by collecting information about node movements in a simulation, generates statistics by which different mobility models can be compared. These statistics can also be used to reason about how different characteristics of the mobility models affect routing performance, and the function of other higher layer protocols. We presented our analysis of the three mobility models we added to OMNeT++ plus random waypoint mobility which was already available.

We plan to implement additional mobility models and are already working on a group mobility model based on Village mobility (but with roving "city centers"), and recent work in the area of mobility models such as the Working Day Movement Model [7] suggest others we should implement. Addi-

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647

	Random Waypoint	Zebra	Village	Levy (α=0.5)	Levy (α=1.5)
Num. of Neighbors	0.26	0.40	3.89	0.19	0.17
Contacts/hr	48.89	21.15	134.86	26.58	10.41
Contact Duration	19.77 sec	69.90 sec	104.14 sec	25.62 sec	59.27 sec
Unique Neighbors	48.99	47.40	49.00	46.61	10.92
Partition Size	2.19	2.33	4.42	2.09	2.09
Num. of Partitions	43.80	41.20	24.70	45.50	45.97
Num of Discon. Nodes	38.60	34.53	17.00	41.40	42.26
Relative Mobility	11.40	6.61	5.68	10.00	5.32

Table 1: Comparative statistics. All entries are averages of values for all nodes over the entire simulation.

tionally, we will need to implement a suite of popular DTN routing protocols, such as PRoPHET [17]. Most research in the area of delay tolerant networking relies on simulation to validate ideas. Since the validity of simulation depends on the realism of the underlying radio model, we are also experimenting with different radio modules for OMNeT++. Finally, no DTN simulation suite would be complete without an implementation of the DNTRG's Bundle Protocol [25] and this is also left as future work.

Acknowledgements

The authors would like to thank the Center for Excellence in Distributed Global Environments for providing research facilities and the collaborative environment. This research was funded in part by the DoD. The views and conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

7. REFERENCES

[1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. Glomosim: A scalable network simulation environment. *UCLA Computer Science Department Technical Report*, 990027, 1999.

[2] R. Baumann, F. Legendre, and P. Sommer. Generic mobility simulation framework (GMSF). In *Proc. of MobilityModels*, pages 49–56, 2008.

[3] B. Burns, O. Brock, and B. Levine. Mora routing and capacity building in disruption-tolerant networks. *Ad Hoc Networks*, 6(4):600–620, 2008.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Comm. and Mobile Computing*, 2(5):483–502, 2002.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. on Mobile Computing*, pages 606–620, 2007.

[6] C. de Waal and M. Gerharz. BonnMotion: A mobility scenario generation and analysis tool. `http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewall/BonnMotion`, 2003.

[7] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *Proc. of MobilityModels*, pages 33–40, 2008.

[8] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks dtmns: Controlled flooding in sparse mobile networks. In *Proc. of NETWORKING*, pages 1180–1192, May 2005.

[9] S. Hong, I. Rhee, S. Kim., K. Lee, and S. Chong. Routing performance analysis of human-driven delay tolerant networks using the truncated levy walk model. In *Proc. of MobilityModels*, pages 25–32, 2008.

[10] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *ACM SIGCOMM Computer Comm. Review*, 34(4):145–158, October 2004.

[11] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proc. of MobiCom*, pages 217–229, 2003.

[12] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proc.of MobiCom*, pages 195–206, 1999.

[13] D. Johnson, J. Broch, Y. Hu, J. Jetcheva, and D. Maltz. The cmu monarch project's wireless and mobility extensions to ns. *Proc. of IETF*, 1998.

[14] E. Jones, L. Li, J. Schmidtke, and P. Ward. Practical routing in delay-tolerant networks. *IEEE Trans. on Mobile Computing*, pages 943–959, 2007.

[15] P. Juang, H. Oki, W. Want, M. Maronosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *ACM SIGPLAN Notices*, 37(10):96–107, October 2002.

[16] A. Keranen and J. Ott. Increasing reality for dtn protocol simulations. *Networking Laboratory, Helsinki University of Technology, Tech. Rep*, 2007.

[17] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *Proc. of the 1^{st} Int'l. Wkshp. on Service Assurance with Partial and Intermittent Resources*, pages 239–254, 2004.

[18] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proc. of WOWMOM*, pages 183–189, 2005.

[19] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proc. of REALMAN*, pages 31–38. ACM New York, NY, USA, 2006.

[20] J. Ott, D. Kutscher, and C. Dwertmann. Integrating DTN and MANET routing. In *Proc. of CHANTS*, pages 221–228, 2006.

[21] P. Pathirana, N. Bulusu, A. Savkin, and S. Jha. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Trans. on Mobile Computing*, 4(3):285–296, May–June 2005.

[22] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78–83, January 2004.

[23] A. Petz and J. Enderle. Statistics module for OMNeT++. `http://users.ece.utexas.edu/~petz/statistics.html`, 2008.

[24] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility: Do humans walk like monkeys? Technical report, North Carolina State University, 2007.

[25] K. Scott and S. Burleigh. Bundle protocol specification. *IETF Draft, draft-irtf-dtnrgbundle-spec-01.txt, October*, 2003.

[26] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[27] A. Varga et al. The omnet++ discrete event simulation system. In *Proc. of ESM*, 2001.

[28] B. Vellambi, R. Subramanian, F. Fekri, and M. Ammar. Reliable and efficient message delivery in delay tolerant networks using rateless codes. In *Proc. of MobiOpp*, pages 91–98, 2007.

[29] J. Widmer and J. Le Boudec. Network coding for efficient communication in extreme networks. In *Proc. of WDTN*, pages 284–291, 2005.

[30] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proc. of INFOCOM*, pages 1312–1321, 2003.

[31] X. Zhang, J. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *Proc. of MobiCom*, pages 195–206, 2007.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5647
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5647