MONGOOSE: a MObility sceNario Generation tOOl for Structured Environments

Adriano Galati[*]
Distributed Systems and
Services Research Group
School of Computing
University of Leeds
LS2 9JT, UK
a.galati@leeds.ac.uk

Karim Djemame[†]
Distributed Systems and
Services Research Group
School of Computing
University of Leeds
LS2 9JT, UK
k.djemame@leeds.ac.uk

Chris Greenhalgh[‡]
Mixed Reality Lab
School of Computer Science
University of Nottingham
NG8 1BB, UK
cmg@cs.nott.ac.uk

ABSTRACT

Mobility models play a key role in the evaluation of wireless network protocols and applications. Simulation results obtained with an unrealistic model may not reflect the true performance of a system in real environments. Besides, different environments may be characterized by different patterns of mobility which require simulation scenarios to be supported by suitable mobility models. The choice of the mobility model and its parameters has significant impact on the result of the simulations. In this paper we present MONGOOSE (MObility sceNario Generation tOOl for Structured Environments), a highly flexible and easily extendable software tool for the simulation of mobility scenarios in structured environments such as shopping malls, urban areas, museums, schools, hospitals, music festivals, amusement parks, stadiums, airports. Moreover, new mobility models can be easily added to the list of those already provided and some of their mobility parameters and simulation playground can be described without programming requirements. Here, we use MONGOOSE to show the impact of four different mobility models on the performance of two well-known delay tolerant routing protocols, Epidemic and Prophet.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*Environments*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Store and forward networks, Wireless communication*

General Terms

Design; Experimentation; Measurement; Performance

Keywords

mobility model, scenario generation software, delay tolerant mobile ad-hoc networks

1. INTRODUCTION

Evaluation of performance of applications and protocols in mobile ad-hoc delay tolerant networks is usually based on simulations. Designing realistic mobility models is one of the most critical and difficult aspects of the simulations of protocols and applications in mobile environments. Even though, real traces identifying user mobility would be preferable for evaluating the characteristics of mobile ad hoc network protocols, it is very difficult to trace node movement in real large-scale mobile environments. Although random movement models generate traces that do not reflect real world observations, synthetic traces are very often used when simulating mobile networks. That is due to many reasons. First, even though there is a repository of public available wireless traces, CRAWDAD [1], the number of available real traces in the public domain is limited and covers a number of specific scenarios. Also, real traces do not provide information such as the distribution of the speed or the density of the hosts, which prevents sensitivity analysis. Moreover, sometimes it may be more useful to have a mathematical model that describes the movement of the nodes in a simulation in order to consider its impact on the design of protocols and applications.

In this paper we propose MONGOOSE, a highly flexible and easily extendable software tool that generates structured mobility traces. MONGOOSE not only allows an easy integration of new mobility models in the list of those already provided by this distribution, but it offers the possibility to incorporate floor plans, obstacles, pathways and boundaries in the simulation playground and to have the number of nodes varying with time in order to create more realistic movements. MONGOOSE generates mobility traces mainly for the OMNeT++ simulator [2, 3]. We define as structured environment, a scenario having definite organization, where people follow characteristic patterns of relationship and mobility. The choice of the mobility model and its parameters has significant impact on the result of the simulations. As proof of concept we present an initial simulation study of the impact of four different mobility models, which simulate a shopping mall scenario, on the performance of two well-known delay tolerant routing protocols, Epidemic [4] and Prophet [5]. MONGOOSE extends our previous

---

[*] Dr. Adriano Galati
[†] Dr. Karjm Djemame
[‡] Prof. Chris Greenhalgh

OMNeT Workshop 2013, March 05-07

DOI 10.4108/icst.simutools.2013.251606

work [6] where we propose a mobility model for shopping mall environments. We have been considering applications in shopping mall environments as they offer all of the elements required to build large-scale delay tolerant mobile ad-hoc networks. Sometimes such networks can be a better solution with respect to traditional networks which could be more expensive, involve installation issues, incur customer cost factors and particular policy restrictions, and may be inapt for wireless people-centric networks where services are established on the fly.

The paper is structured as follows: in Section 2 we present MONGOOSE, our mobility scenario generation tool; in Section 3 we show the simulation study; in Section 4 we introduce the tools that are related to our application. Finally, Section 5 concludes the paper and points out future developments.

2. MONGOOSE SOFTWARE TOOL

We have developed a tool that generates mobility models for structured environments and a number of other random based mobility models. Our mobility scenario generation tool is a Java application. It provides certain features to the user to model and analyse mobility in structured environments. The main features of the software developed are the following:

- It creates structured scenarios and allows further mobility models to be easily hooked.
- Mobility traces for some traditional random mobility models can also be produced.
- The plan structure can be described without programming requirements (see Figure 1).
- Fine-grained movement traces for shopping mall scenarios as well as for other different structured environments can be generated.
- The generated mobility traces are compatible with the OMNeT++ simulator [2, 3], one of the most popular discrete event network simulation framework in the mobile ad hoc network research community.
- Easy to use: starting the program without or with incomplete command line parameters prints a detailed help message.

For a supported mobility model it requires input parameters appropriate to the model and outputs a list of node trajectories.

2.1 Requirements and Design

To design MONGOOSE we have considered two components: a spatial component and a temporal component. The spatial component describes where the mobile entity is moving, and the temporal component describes when an entity is moving and at which speed. To define the environment that we would like to reproduce, which is made of boundaries, obstacles, walls, paths, intersections and restrictions of the simulated world an SVG application can be used. SVG (Scalable Vector Graphic) [7] is a language for describing two-dimensional graphics and graphical applications in XML. The SVG specification is an open standard that has been under development by the W3C (World Wide Web Consortium). MONGOOSE has been tested with Inkscape 0.46 [8], an open source SVG graphics editor released under the GNU GPL (see Figure 1). Besides, MONGOOSE requires more parameters to build the simulated world: the simulation time, the random seed, the higher and lower speed of the nodes, and pause between two successive movements. These parameter values are used by MONGOOSE to generate different mobility scenarios. Our tool provides three configurations to generate several mobility models, the `SimplestRWP`, the `RandomWayPoint` and the `StructuredMotion`, which are described in the next sections.

MONGOOSE allows an easy integration of new mobility models. MM looks in its list of implemented models and gets a class object which represents a mobility model. A public member function `go` determines the class to run which is the configuration specified on the command line (see Section 2.3) and passes the parameters to the model. Reflection is used to look in its list of implemented models. MONGOOSE is available upon request to the research community.

2.2 Usage

The application starts through the command `MM`. The syntax is the following:

```
MM <output file> <application> <plan> <parameters>
```

`<application>` identifies one of the three configurations we have implemented to generate mobility scenarios. MONGOOSE takes as input an svg file `<plan>`, which provides a plan, some cumulative distribution functions, which assign time to all of the activities in the simulation playground, and some information to model the scenario. It also takes `<application parameters>` on the command line which define simulation time, speed range and pause time of the nodes involved. Important parameters which could be also used with all the models are the following: the random seed with `-R`, which can be optional as it can be automatically chosen, the maximum and minimum speed in metres per second respectively with `-h` and `-l` and pause time with `-p`, the scenario duration (in seconds) with `-d` and the `-i` parameter specifying how many additional seconds at the beginning of the scenario should be skipped. Cutting off the initial phase is an important feature for traditional mobility models. It has been observed that with the Random Way-Point model [9, 10], nodes have a higher probability of being near the center of the simulation area, while they are initially uniformly distributed over the simulation area. Therefore, `-i` has a high default value of 3600 seconds.

The movement traces must be saved into a file by means of the option ”`-f <output file>`. The scenario is saved in two files: one with the suffix ”.params” containing the complete set of parameters used for the simulation, and the second with the suffix ”.movements.gz” containing the movement data.

2.3 Configurations

`StructuredMotion`

This configuration is able to model structured scenarios. Here we refer to our Shopping Mall mobility model [6] as structured scenario, which will help to describe better this configuration. In this scenario we consider two groups of people expressing different mobility patterns, internal and external nodes which represent sellers and customers respectively. The configuration to generate structured mobility models consists of several components. The first component is the playground in which shapes and sizes of the structures, such as rooms, buildings, pathways and obstacles, are defined and sellers are placed (see Figure 1). It presents the plan and settings saved in svg format and handed to MONGOOSE. We introduce nodes in the playground by drawing black dots which are initially positioned in their workplace. The red dots identify fixed nodes. Our model can handle any arbitrary shape and position of structures which allow us to model a wide variety of real-world topographies.

The second component is the movement backbone which is a graph representing the pathways along which mobile nodes move to go from one place to the other. Mall common spaces, intersections and entrances are respectively identified by $V : [1..n]$ and $\wedge$. These guide nodes through the mall. $\wedge$ identifies the starting and ending point of customers outside the mall. Figure 2 shows the pathway graph of the shopping mall taken into account in Figure 1. They move from one location to another in the mall by choosing the shortest path along this movement backbone. To accomplish this we used the Dijkstra's algorithm on a graph where the mall interceptions $V : [1..n]$, shops and stores are the vertices, and the corridors, streets and galleries are the edges (see Figure 2). In shops nodes move with a Random WayPoint mobility model [9].

The third component is the destination selection. We assign internals and externals' attraction levels (i.e. `ea:` "externals attraction level" and `ia:` "internals attraction level" followed by an integer number) to each room of the mall that would influence the choice of individuals in going to one shop rather than another; higher values being the more attractive.

Finally, the fourth component is a set of cumulative distribution functions that dictate how nodes move within the shopping mall. The model is stochastic: it determines each nodes' movement by the random sampling of the provided cumulative distributions. We feed MONGOOSE with the following cumulative distribution functions characterizing:

- the time spent by internals (sellers) within and out of their workplace;
- the time spent by externals (customers) within shops and mall;
- and the externals' inter-arrival time.

The statement lines describing these processes are embedded in the svg file and parsed by MONGOOSE. In Figure 1 these are at the bottom but they can be placed anywhere in the document. The statement lines (1) and (2) describe such cumulative distributions. They are composed of two parts: the right hand side, describing a cumulative distribution function, and the left hand side, describing the purpose of the formula. Their syntax is the following:

$$\underbrace{iat}_{\substack{\text{inter-arrival}\\ \text{time}}} : \underbrace{F}_{\substack{\text{cum.}\\ \text{distr.}}} \quad \underbrace{(\alpha,\ \beta,\ ...)}_{\substack{\text{parameters of}\\ \text{the distribution}}} \tag{1}$$

$$\begin{aligned} &stay[(external, [sim\_area|sub\_area])|\\ &\quad(internal, [sub\_in|sub\_out])] : \underbrace{F}_{\substack{\text{cum.}\\ \text{dist.}}} \quad \underbrace{(\alpha,\ \beta,\ ...)}_{\substack{\text{parameters of}\\ \text{the distribution}}} \end{aligned} \tag{2}$$

MONGOOSE supports five cumulative distribution functions: exponential, gamma, lognormal, weibull and linear:

- Exponential: `exp(rate=...)`

Figure 1: Plan of the shopping mall drawn using Inkscape

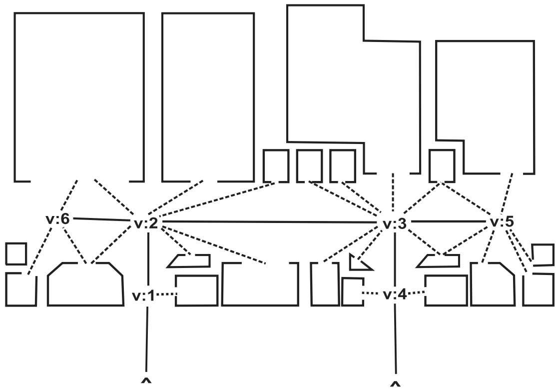

Figure 2: Pathway graph along which mobile nodes move in the shopping mall to go from one shop to the other.

- Lognormal: `lnorm(meanlog=..., sdlog=...)`
- WeibullL: `weibull(shape=..., scale=...)`
- Gamma: `gamma(shape=..., scale=...)`
- Linear: `linear([a<=y<=b, slope=..., \ intercept=...;]) where a>=0 and b<=1`

It is also possible to consider a system of linear distributions and contiguous codomains. The line (1) describes the inter-arrival time of externals to the simulation playground. The left hand side of line (2) tells the mobility model generator the purpose of the function alongside. As you can see on the bottom of Figure 1, `StructuredMotion` needs four statement lines following the syntax in (2) which mean:

- `stay(external, sim_area)`: how much time externals usually spend in the simulation area (i.e. mall);
- `stay(external, sub_area)`: how much time externals usually spend in a sub area (i.e. shop of the mall);
- `stay(internal, sub_in)`: how long internals usually stay within their local area (i.e. sellers within their workplace);

- `stay(internal, sub_out)`: how long internals usually stay out of their local area.

`SimplestRWP`

This configuration can generate the typical Random Way-Point [9, 10] as well as the Random Walk mobility model [11]. Because of their simplicity, they became soon 'benchmark' mobility models to evaluate MANET routing protocols. The Random Way-Point mobility model is a derivative of the Random Walk model. It is based on random directions and speeds like the Random Walk, but it also includes pause times between changes in destination and speed. `SimplestRWP` reproduces these two traditional models within a certain area. The generation of a Random Walk rather than a Random Way-Point mobility model depends on the parameter `-p`, which indicates the maximum pause time. If `-p = 0` this will result in a Random Walk model. The following example command line generates a Random Walk mobility model (`-p = 0`). This configuration does not take any distribution function. The only nodes in play are the ones drawn as black dots on the svg plan. This means that in this mobility model the number of nodes is constant.

`RandomWayPoint`

As above mentioned, the Random Way-Point is a simple mobility model based on random direction, speeds and pauses. The Random Way-Point Model was first proposed by Johnson and Maltz [9] and it is widely used in simulations of mobile networks [12, 13, 14, 15]. This configuration supports a variation of the classical Random Way-Point and Random Walk models in which nodes enter and leave the playground over time. Specifically, nodes arrive at specific points of the playground with a certain frequency; after that, they follow a Random Walk or Random Way-Point mobility model; and finally, when their time expires, they move to the closest exit and get out of the playground. In this way, the specific nodes involved in the scenario varies with the time although the number is reasonably constant over time. As for the previous models, the generation of a Random Walk mobility model rather than a Random Way-Point mobility model depends on the `-p` pause parameter. To generate the Random Walk and the Random Way-Point mobility models we need to provide the scenario generator with the following cumulative distribution functions:

- the time spent by nodes in the playground,
- and the nodes' inter-arrival time.

2.4 External Libraries

MONGOOSE also depends on two external Java libraries, *SSJ* and *Apache Commons Mathematics*. *SSJ* is a library for stochastic simulation developed in the Département d'Informatique et de Recherche Opérationnelle (DIRO), at the Université de Montréal [16]. It provides facilities for generating uniform and nonuniform random variates, computing different measures related to probability distributions, performing goodness-of-fit tests, applying quasi-Monte Carlo methods, collecting statistics, and programming discrete-event simulations with both events and processes. *Apache Commons Mathematics* is a library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language [17].

3. SIMULATION STUDY

Simulations of mobile networks strongly rely on mobility models because they have a major influence on the performance of wireless network protocols and applications. Simulation results obtained with an unrealistic model may not reflect the true performance of a system in real environments. Besides, different environments may be characterized by different patterns of mobility which require simulation scenarios to be supported by suitable mobility models. The choice of the mobility model and its parameters has significant impact on the result of the simulations. As proof of concept we test and evaluate Epidemic [4] and Prophet [5] routing protocols, two well-known routing protocols employed in Delay Tolerant Networks, with our Shopping Mall [6] and three other mobility models. We intend to reproduce a shopping mall scenario based on four different mobility models: Random Walk, Random Way-Point with inter-arrival time, Random Walk with inter-arrival time and the Shopping Mall mobility model. We include the Random Walk and Random Way-Point models with inter-arrival time to allow us to separate out the effects of a changing customer population from the additional structure of the Shopping Mall mobility model. We chose Epidemic and Prophet because they are often used as benchmark routing protocols in simulations carried out by the delay tolerant network research community to simulate several scenarios for mobile networks [9, 11, 18, 19]. In addition, Epidemic provides a theoretical upper bound in terms of delivery ratio when the buffer size is infinite. Epidemic is a resource hungry protocol because it does not use any knowledge of the system to forward messages. Prophet, in contrast, tries to exploit the non-randomness of individuals' encounters by maintaining a list of delivery probabilities for known destinations. This presumes that human mobility is often goal-oriented and that encounters could be predictable.

To run our simulations we adopted Mixim [20], a simulation framework for wireless and mobile networks which uses the OMNeT++ simulation engine [3]. Mixim supports mobile and wireless simulations and offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols. OMNeT++ can carry out large-scale simulations, only limited by the virtual memory capacity of the computer used. Because of the amount of data we processed and the number of simulations we performed, we made extensive use of the HPC (High Performance Computing) of the University of Nottingham.

3.1 Simulation Scenarios

The movement traces of the simulated hosts were generated using MONGOOSE. All the scenarios have in common 45 sellers and a simulation area of $10,880m^2$. We were very precise in drawing the shopping mall as we have its original plan. The scale of the simulation playground is 2:1 (i.e. 2 pixels = 1 metre). We set the attraction level of each shop equal to 1 so that nodes have the same probability to move to any shop. These attraction level settings are acceptable for small scale environments like the shopping mall we are considering but it may not be suitable for large scale environments. We employed four distinct mobility models characterized by different degrees of freedom for nodes to move with respect to time and space. Decreasing spatial mobility and speed and increasing pause time reduces the degree of freedom of nodes. Fixed nodes are not considered for these simulations. The number of customers varied according to the cumulative inter-arrival time distribution.

The first scenario is the Shopping Mall mobility model [6]. This mobility model considers obstacles as well as shops, stores, rooms, walls, and paths. The modelled mall can be seen in Figure 1. A command line to reproduce a shopping mall scenario is the following:

```
MM -f scenario StructuredMotion drawing.svg \
         -d 43200 -i 3600 -h 1.65 -l 1.15 -p 2  (3)
```

The speed and pause time of the nodes are randomly generated according to a uniform distribution respectively between [1.65 - 1.15]m/s and [0 - 2]s, based on [21, 22] where the walking speed of people in urban areas is analyzed.

Furthermore, as described in the previous section, MONGOOSE needs five cumulative distribution functions to generate this scenario. These distributions are extracted from our data set of real world contact traces [6, 23, 24, 25, 26]. Following ethical principals, an anonymised version of our data has been made publicly available to other research groups on CRAWDAD [1] (i.e. with MAC addresses mapped to synthetic IDs with only local relevance). The distributions together with their respective parameters are the following:

```
iat:  exp(rate=1.771467e-03)                        (4)
stay(externals,sim_area):
    weibull(shape=9.3573e-01,scale=2.5797e+03)  (5)
stay(externals,sub_area):
    weibull(shape=1.0028e+00,scale=3.0595e+02)  (6)
stay(internals,sub_in):
    lnorm(meanlog=7.08653949,sdlog=1.87655822)  (7)
stay(internals,sub_out):
    lnorm(meanlog=6.50409971,sdlog=0.51007676)  (8)
```

The second and third scenarios are a Random Way-Point and Random Walk mobility model with inter-arrival time respectively. Command lines that generate twelve hours of each scenario are the following:

```
MM -f scenario RandomWayPoint RWP.svg \
   -d 43200 -i 3600 -h 1.65 -l 1.15 -p 2      (9)

MM -f scenario RandomWayPoint RWP.svg \
   -d 43200 -i 3600 -h 1.65 -l 1.65 -p 0     (10)
```

The command line (9) uses a Random Way-Point model with speed in the range $[1.65 - 1.15]m/s$ and a maximum pause time of $2s$. The command line (10) generates a Random Walk with constant speed of $1.65m/s$. A node's speed in this scenario is equal to the highest permitted speed in the previous one. These two scenarios make use of two cumulative distributions to describe the customers' inter-arrival time (4) and their staying time in the mall (5). In the first three scenarios the number of customers in the simulation is not constant, but varies according to the distributions.

In the last scenario, which represents a Random Walk mobility model, 225 nodes were always present This is the sum of 45 sellers and the arithmetic mean calculated on a sample of one hundred recordings of the number of customers in the shopping centre after two hours (after two hours the system is in a steady state; the average number of customers in the mall falls within a certain range, namely the probability of customers leaving the mall is equal to their arrivals) following the inter-arrival time cumulative distribution function (4) and a random number generator. Notice that this is the number of individuals carrying an active Bluetooth device; people might turn Bluetooth appliances off if not needed or to save battery power. The following command line (11) generates this last scenario with constant speed of $1.65m/s$ and duration equal to 12 hours.

```
MM -f scenario SimplestRWP Simple.svg \
   -d 43200 -i 3600 -h 1.65 -l 1.65 -p 0     (11)
```

In the last three mobility models nodes move without taking into account the plan of the mall, e.g. walls, corridors and obstacles. This means that nodes can move everywhere within the boundaries as in an open space.

3.2 Network Setting

We consider the mobile scenario at the network level and do not consider issues related to Physical and MAC layers such as packet loss, collision or signal fading and do not deal with retransmission of packets. We assume that two devices can simply transmit messages when they are in radio range. Consequently, we do not model retransmission of packets. Here, we provide a basic comparison of routing protocols with a reasonable approximation and slightly optimistic results. Nonetheless, this is sufficient for the purpose of this study. With respect to the radio technology, we also assumed a free space propagation model with all the nodes having a transmission range of 30 metres and the use of omnidirectional antennas. The retransmission interval was 134 seconds. These settings are the same as those used by the Bluetooth appliances of our smartphones employed in the field trial to collect contact data in the shopping mall [6, 24, 25, 26].

We assumed a buffer size equal to 100 slots, i.e. whose capacity is 10% of the number of messages sent in the network. We assume that each host is able to store one message per slot. They use summary vectors that index the list of messages stored at each node to mutually find out which messages can be obtained from each other. Each message is identified by a unique message identifier. No messages were sent for the first 7200 seconds, in order to allow the simulation scenario to converge to a steady state, for example, a typical number of customers in the mall. We evaluated the performance of each routing protocol by sending 1000 messages over a simulation time equal to 2400 seconds. The minimum interval between the generation of two subsequent messages is equal to 0.1 seconds, as long as the chosen recipient node is different from the sender. In other words, the generation of all the messages will take at least 100 seconds. In our network scenario the generation of messages with such a frequency produces a certain network load which will stress the network itself and allows network protocols to show better the capability of network protocols in routing data. Network load impacts on the performance of routing protocols as it might give rise to data loss at intermediate nodes when they are overloaded. The sender and recipient of each message were randomly chosen among all of the nodes in the mall following a uniform distribution. Thus, it might happen that they are about to leave the mall and unable to deliver their messages. This choice may be unrealistic, however, it is clearly less optimistic than assuming that communication happens only between people just arrived at the shopping mall. The main problem in designing realistic traffic models for delay tolerant networks is the lack of real data for validating it, especially for shopping mall environments. We run equal number of simulations with each combination

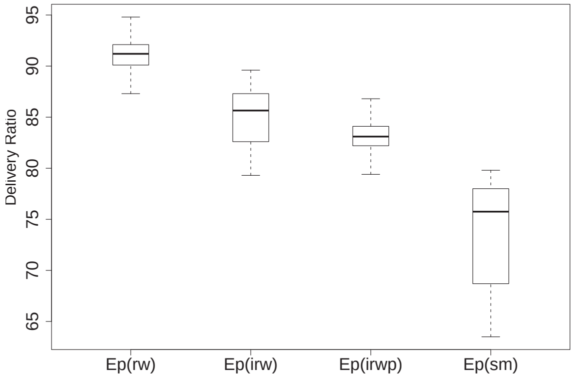

Figure 3: Boxplot of the delivery ratio for the Epidemic routing protocol with the Random Walk (`rw`), Random Walk with inter-arrival time (`irw`), Random Way-Point with inter-arrival time (`irwp`) and the Shopping Mall (`sm`) mobility model.

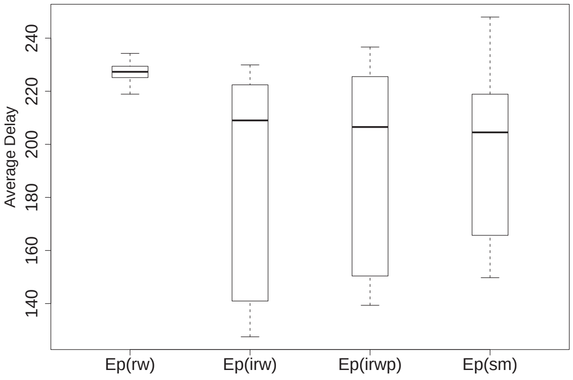

Figure 4: Boxplot of the average delay for the Epidemic routing protocol with the Random Walk (`rw`), Random Walk with inter-arrival time (`irw`), Random Way-Point with inter-arrival time (`irwp`) and the Shopping Mall (`sm`) mobility model.

of customers and sellers as senders and receivers: i.e. the number of runs was 50 for each combination of each scenario. This was sufficient to determine a 95% confidence interval using a t-distribution.

3.3 Simulation Results

Here, we analyze the impact of different mobility models on the performance of the Epidemic and Prophet routing protocols. The mobility models are identified in the plots with the following order and with the respective labels: `rw`: Random Walk, `irw`: Random Walk with Inter-arrival time, `irwp`: Random Way-Point with Inter-arrival time, `sm`: Shopping Mall.

These are in order of decreasing degree of freedom of nodes: `rw` and `irw` use a constant (maximum) node speed without pauses, `irwp` uses speeds between an upper and lower bound and non-zero pause time while `sm` divides the simulation area in smaller spaces representing the shopping mall. In all cases, the maximum node speed is $1.65m/s$ while `irwp` and `sm` have a possible minimum speed of $1.15m/s$ and a random pause time of up to $2s$ after each movement (see Section 3.1).

Initially we run simulations distinguishing between customers and sellers to be selected as sender and recipient, namely, from customer to customer, customer to seller, seller to customer and seller to seller. Here we unify these results and show them together in a single plot. The first two plots in Figure 3 and 4 show respectively the delivery ratio and the average delay using the Epidemic protocol. In Figure 3 we observe that the delivery ratio decreases with the decreasing degree of freedom of the scenarios. This is in accordance with Grossglauser and Tse [27] who claim that mobility increases the capacity of ad hoc wireless networks. Their results suggest that delay tolerant applications can take advantage of node mobility to significantly increase the throughput of such networks. As regards the average delay of the message to reach the final destination, Epidemic has a slightly higher mean with the traditional Random Walk mobility model while the other three mobility models produce roughly the same mean but with large range of minimum delay. This is because in the Random Walk case all of the nodes are independent and identically distributed in the mall area and so have the same probability for their messages to reach any recipient with the same average delay.

Like Epidemic, in Figure 5 Prophet also shows decreasing mean delivery ratio with scenario, with the exception of the Shopping Mall mobility model, where it performs better than `irw` and `irwp`. This is explained by Prophet's exploitation of the non-randomness of individuals' encounters. Human mobility and encounters in structured environments like shopping malls is likely to be more predictable than in unconstrained scenarios. Recall that the `rw` model does not consider inter-arrival times which are likely to negatively impact the outcome of the simulations. Figure 6 shows the average delay for delivering messages with Prophet. Here, `sm` has the longest average delay.

4. RELATED WORK

There are a wide variety of mobility trace generators that have been proposed to produce mobility models for the evaluation of protocols in delay tolerant mobile ad-hoc networks, ranging from the very simple to the very complex. However, no one supports mobility models for structured environments such as shopping malls, urban areas, museums, schools, hospitals, music festivals, amusement parks, stadiums and airports. The Communication Systems group at the Institute of Computer Science 4 of the University of Bonn, in Germany, has developed BonnMotion [28], a Java software tool which creates and analyses several mobility scenarios for the investigation of mobile ad hoc network characteristics. Bohacek et al. [29] proposed the UDel Models, a suite of simulation tools for realistic simulation of mobile wireless networks in urban area. Unlike many other mobility modelling efforts, the mobility model they provide is based on data collected by the US Bureau of Labor Statistics which produces coarse location information. This model focuses

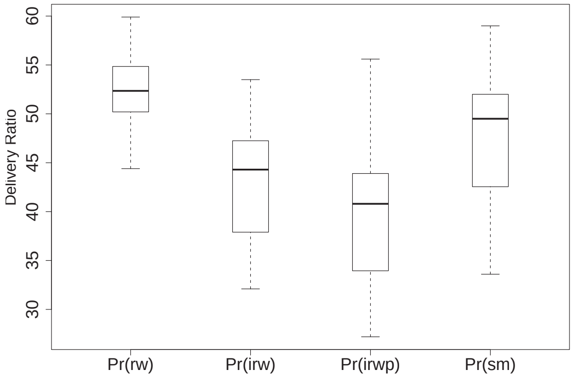

Figure 5: Boxplot of the delivery ratio for the Prophet routing protocol with the Random Walk (`rw`), Random Walk with inter-arrival time (`irw`), Random Way-Point with inter-arrival time (`irwp`) and the Shopping Mall (`sm`) mobility model.

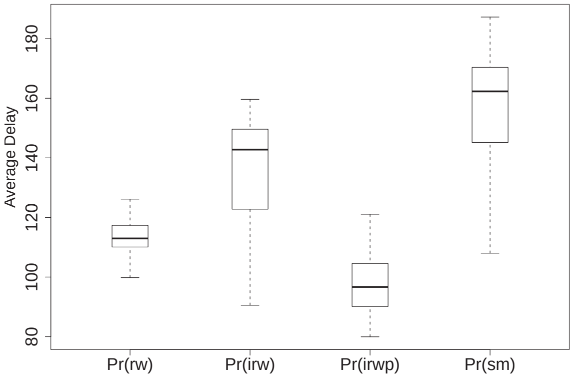

Figure 6: Boxplot of the average delay for the Prophet routing protocol with the Random Walk (`rw`), Random Walk with inter-arrival time (`irw`), Random Way-Point with inter-arrival time (`irwp`) and the Shopping Mall (`sm`) mobility model.

on a typical day-cycle of a working person commuting between home and office with possible activities during breaks at work. Such a model considers activities during working time, i.e. shopping, consisting of going to a place where to wait still or randomly walk for a certain time before going back to the workplace. It is evident that mobility during such activities do not reproduce reality. This lacks submodules to model mobility for different activities in addition to working in office environment. In this case, MONGOOSE can be used to generate mobility traces for the UDel submodules. Helgason et al. [30] presented a module for the OMNeT++ simulator [2, 3] and the Mobility Framework [31] for dynamically creating and destroying nodes during the course of a simulation, modules that implement node mobility or node contacts from trace files and a toolbox of scripts for mobility generation and conversion of output from external mobility generators. Petz et al. [32] developed a suite of mobility models, Zebra Mobility, Village Mobility, and Levy Walk Mobility, in OMNeT++ that specifically target delay tolerant networks. Koberstein et al. [33] discussed a specialized type of graph-based mobility model and novel methods built upon the OMNeT++ simulator. Their graph-based mobility model is designed to resemble probabilistic node movements according to real world node paths like the ones induced by road grids.

5. CONCLUSIONS

In this paper we have presented MONGOOSE, our mobility model generation tool, which produces mobility traces for the OMNeT++ simulator. MONGOOSE generates fine-grained mobility traces for structured scenarios such as shopping malls, urban areas, museums, schools, hospitals, music festivals, amusement parks, stadiums and airports, as well as for some traditional random based mobility models. It can also generate mobility traces for two groups of nodes, internals and externals, which express different mobility patterns. Given proper parameters, our mobility model generator can produce different kind of scenarios. MONGOOSE produces structured scenarios for specific areas and their applicability is limited to such environments. Besides, it allows further mobility models to be easily plugged-in. Moreover, MONGOOSE reduces programming requirements of users to build the plan structure as it can be drawn by means of SVG graphics editors. We have also shown that the choice of a mobility model affects the performance of both routing protocols, Epidemic and Prophet. In the future we would like to add further mobility models in the list of those already provided by this distribution and to provide related parameters to generate more realistic mobility scenarios. We would also like to improve MONGOOSE by considering group relationships and more subpopulations expressing different mobility patterns.

6. ACKNOWLEDGMENTS

We would like to thank Dr. Mirco Musolesi for providing us the code of the Epidemic and Prophet routing protocols. We are also thankful to the High Performance Computing group of the University of Nottingham for allowing us to run large-scale simulations.

7. REFERENCES

[1] D. Kotz and T. Henderson. CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth. *IEEE Pervasive Computing*, pages 4(4):12–14, 2005.

[2] A. Varga. The OMNeT++ Discrete Event Simulation System. *Proceedings of the European Simulation Multiconference (ESM'2001)*, June 2001.

[3] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Simutools '08*, (60):1–10, 2008.

[4] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. *Technical*

Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[5] A. Lindgren, A. Doria, and O. Schelén. Probabilistic Routing in Intermittently Connected Networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[6] A. Galati, K. Djemame, and C. Greenhalgh. A Mobility Model for Shopping Mall Environments Founded on Real Traces. *Springer/Tsinghua University Press journal, Networking Science*, DOI: 10.1007/s13119-012-0011-1, 2012.

[7] Scalable Vector Graphics. [online] Available: `http://www.w3.org/Graphics/SVG`.

[8] Inkscape: Open source vector graphics editor. [online] Available: `http://www.inkscape.org`.

[9] D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing, T. Imielinski and H. Korth, Eds., Ch. 5, Kluwer*, pages 153–181, 1996.

[10] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. *INFOCOM*, 2003.

[11] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad-hoc Network Research. *Wireless Communication and Mobile Computing Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, pages 2(5):483–502, 2002.

[12] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad hoc network routing protocols. *In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–97, 1998.

[13] C. Chiang and M. Gerla. On-demand multicast in mobile wireless networks. *In Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 1998.

[14] J.J. Garcia-Luna-Aceves and M. Spohn. Source-tree routing in wireless networks. *In Proceedings of the 7th International Conference on Network Protocols (ICNP)*, 1999.

[15] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Routing protocols for mobile ad-hoc networks - a comparative performance analysis. *In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 195–206, 1999.

[16] SSJ: Stochastic Simulation in Java. [online] Available: `http://www.iro.umontreal.ca/~simardr/ssj`.

[17] ApacheCommons-Math: The Apache Commons Mathematics Library. [online] Available: `http://commons.apache.org/math`.

[18] M. I. M. Saad and Z. A. Zukarnain. Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol. *European Journal of Scientific Research*, 32(4):444–454, 2009.

[19] B. T. M. Mukarram, A. Mostafa, and Z. Ellen. Message Ferry Route Design for Sparse Ad-hoc Networks with Mobile Nodes. *in Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06*, pages 37–48, 2006.

[20] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. Klein Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating Wireless and Mobile Networks in OMNeT++, the MiXiM Vision. *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Simutools '08*, pages 71:1–71:8, 2008.

[21] R. L. Knoblauch M. T. Pietrucha and M. Nitzburg. Field Studies of Pedestrian Walking Speed and Start-up Time. *In Transportation Research Record 1538, TRB, National Research Council*, pages 27–38, 1996.

[22] N. Carey. Establishing Pedestrian Walking Speeds. *Portland State University*, 2005.

[23] Bluetooth contact traces in Shopping Mall environments. [online] Available: `http://crawdad.org/nottingham/mall`.

[24] A. Galati, K. Djemame, and C. Greenhalgh. Opportunistic Forwarding Throughout Customers or Sellers in Shopping Mall Environments. *in the Proceedings of the International Conference IFIP Wireless Days 2012, Dublin*.

[25] A. Galati and C. Greenhalgh. Human mobility in shopping mall environments. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, MobiOpp '10, pages 1–7. ACM, 2010.

[26] A. Galati and C. Greenhalgh. Exploring Shopping Mall Environment for Ubiquitous Computing. *In Ubicomp at a Croassroads, Imperial College London*, 2009.

[27] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, 2002.

[28] BonnMotion: A mobility scenario generation and analysis tool. [online] Available: `http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion`.

[29] S. Bohacek, V. Sridhara, and J. Kim. UDel models for simulating urban wireless networks. [online] Available: `http://udelmodels.eecis.udel.edu`.

[30] O. R. Helgason and K. V. Jonsson. Opportunistic networking in OMNeT++. *In Proceedings of 1st international conference on Simulation tools and techniques for communications, networks and systems workshops (Simutools '08), Marseille, France*, March 2008.

[31] W. Drytkiewicz, S. Sroka, V. Handziski, A. Köpke, and H. Karl. A Mobility Framework for OMNeT++. *In Proceedings of 3rd International OMNeT++ Workshop*, January 2003.

[32] A. Petz, J. Enderle, and C. Julien. A Framework for Evaluating DTN Mobility Models. *In Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09), Belgium*, 2009.

[33] K. Jochen, P. Hagen, and L. Norbert. Graph-based mobility model for urban areas fueled with real world datasets. *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, (Simutools'08)*, 8:1–8, 2008.