Large-Scale Synthetic Social Mobile Networks with SWIM

Sokol Kosta, Alessandro Mei, *Member*, *IEEE*, and Julinda Stefa, *Member*, *IEEE*

Abstract—This paper presents small world in motion (SWIM), a new mobility model for ad hoc networking. SWIM is relatively simple, is easily tuned by setting just a few parameters, and generates traces that look real—synthetic traces have the same statistical properties of real traces in terms of intercontact times, contact duration, and frequency among node couples. Furthermore, it generates social behavior among nodes and models networks with complex social communities as the ones observed in the real traces. SWIM shows experimentally and theoretically the presence of the power-law and exponential decay dichotomy of intercontact times, and, most importantly, our experiments show that predicts very accurately the performance of forwarding protocols for PSNs like Epidemic, Delegation, Spray&Wait, and more complex, social-based ones like BUBBLE. Moreover, we propose a methodology to assess protocols on model with a large number of nodes. To the best of our knowledge, this is the first such study. Scaling of mobility models is a fundamental issue, yet never considered in the literature. Thanks to SWIM, here we present the first analysis of the scaling capabilities of Epidemic Forwarding, Delegation Forwarding, Spray&Wait, and BUBBLE.

Index Terms—Mobility model, small world, simulations, forwarding protocols in mobile networks

1 Introduction

Pocket switched networks (PSN), networks of mobile humans carrying short-range communication devices such as smartphones, PDAs, or laptops, have received significant attention from the research community during the last few years. The complexity of these networks derives mostly from the difficulty of predicting human mobility. Much research has been dedicated to the study of real-life experimental data traces [1], [2], [3], [4], [5], [6] so as to compute statistical properties of human mobility and, therefore, of PSNs. These works have mostly focused on intercontacts (time intervals between two consecutive contacts of the same couple of nodes), contact duration, and contact number distributions among node pairs, and have confirmed the complexity and the unpredictability of human mobility. Another large flow of works have been dedicated to uncovering structural properties of PSNs such as the presence of social-based community substructures [7], [8], [5] and to using these properties to design efficient message forwarding [8]. Additionally, in [9] the authors discuss on the limits of experiments based on logging contacts and show how to infer plausible mobility patterns from them.

Also have a large number of works been presented on designing models for human mobility [10], [11], [12], [13], [14], [15], [16], [17]. Most of these works validate their models with real-life data traces available online and unfortunately not very large.

In this work, we present small world in motion (SWIM [17], [18]), a simple mobility model that generates small worlds of mobile humans. The model is very simple to implement and very efficient in simulations. The mobility pattern of the nodes is based on a simple intuition on human mobility: People go more often to places not very far from their home and where they can meet a lot of other people. By implementing this simple rule, SWIM is able to raise social behavior among nodes, a fundamental ingredient of human mobility in real life. We validate the model using four different real traces and compare the distributions of intercontact times, contact durations, and number of contacts between nodes, showing that synthetic data that SWIM generate match very well each of the four real scenarios simulated. The features of SWIM are as follows:

- It is the first model to show—mathematically, not only experimentally—the power-law exponential dichotomy of intercontact times that has been observed in the real-life experiments;
- it generates traces with similar statistical properties (distribution of intercontact times, contact number, and contact durations among couples) and social community structure to well-known, small-scale experimental traces;
- it validates correctly sophisticated protocols based on the social structure of the network such as BUBBLE [8] (as well as Delegation [19], Epidemic [20], and Spray&Wait [21]);
- it is able to generate easily large (small)-scale scenarios, starting from known small (large)-scale ones.

This last feature of SWIM allows us to address the fundamental problem of generating *large scale* synthetic social mobile networks that can be used to assess the performance of forwarding protocols. SWIM-generate larger versions of well-known real-life experiments on human mobility in two different ways—larger number of nodes

- *The authors are with the Department of Computer Science, Sapienza University of Rome, Via Salaria 113-terzo piano, 00198 Rome, Italy. E-mail: {kosta, mei, stefa}@di.uniroma1.it*

Manuscript received 19 Mar. 2012; revised 19 July 2012; accepted 12 Oct. 2012; published online 23 Oct. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2012-03-0140. Digital Object Identifier no. 10.1109/TMC.2012.229.

1536-1233/14/$31.00 © 2014 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

and same network area (the *Manhattan model*), and larger number of nodes and same density (the *Phoenix model*)—and then use these traces to validate the aforementioned forwarding protocols. SWIM is able to extrapolate key properties of human mobility and can be used to understand how protocols scale to larger and larger networks. To the best of our knowledge, this is the first mobility model that addresses this issue and this is the first work that can show reliable performance evaluation of well-known forwarding protocols on large scale networks.

The rest of the paper is organized as follows: Section 2 briefly reports on current work in the field; in Section 3, we discuss the fundamental requirements of a good mobility model; in Sections 3.1, 3.2, and 3.3, we describe the way SWIM operates and mathematically prove the presence of exponentially distributed tail of the intercontact times in SWIM, whereas Section 3.5 describes the methodology used to make it able to scale up. In Section 4, we show experimentally the good matching between statistical properties of SWIM and the real traces, present the experiments related to the enlarged SWIM-scaling scenarios, and show how remarkably similarly Epidemic [20], Delegation [19], Spray&Wait [21], and BUBBLE [8] perform on both the real and synthetic SWIM-generated traces. In Section 5, we show for the first time how these protocols perform on the enlarged scenarios, and give insights on their scaling properties. Section 6 shows how to customize SWIM to generate networks with known community substructure. We lastly conclude with Section 7.

2 Related Work

The problem of designing a mobility model for human mobility is felt as an important one in the community and in the literature. In the last few years, there have been a considerable number of papers on this topic. The work in [22] is one of the first to argue heterogeneous movement of nodes and to present a mobility model, where nodes target a few concentration destination points in the area.

More recently, the model presented in [12] generates movement traces using a model which is similar to a random walk, except that the flight lengths and the pause times in destinations are generated based on Levy Walks—with power-law distribution. In the past, Levy Walks have been shown to approximate well the movements of animals. The model produces intercontact time distributions similar to real-world traces. However, because every node moves independently, the model does not generate any social structure in the network. In [10], [11], the authors present a mobility model based on social network theory that takes in input a social network and discuss the community patterns and groups distribution in geographical terms. They validate their synthetic data with real traces and show a good matching.

The work in [13] presents a new mobility model for clustered networks. Moreover, a closed-form expression for the stationary distribution of node position is given. The model captures the phenomenon of emerging clusters, observed in real partitioned networks, and correlation between the spatial speed distribution and the cluster formation. In [14], the authors present a mobility model that simulates the everyday life of people that go to their workplaces in the morning, spend their day at work, and go back to their homes at evenings. Each one of these scenarios is a simulation per se. The synthetic data they generate match well the distribution of intercontact time and contact durations of real traces. In [15], the authors proposed the SLAW mobility model, which is a modification of the Levy-walk-based model, where the human waypoints are modeled as fractals. The model matches well the intercontact times distribution of the real traces, and predicts quite accurately performance of simple forwarding protocols. Yet, no results are presented in terms of contact duration and contact number distributions and in the structure in communities of the resulting network, and the model seems to be hard to be used in theoretical analysis.

The work of Barabasi et al. [23] studies the trajectory of a very large (100,000) number of anonymized mobile phone users whose position is tracked for a six-month period. They observe that human trajectories show a high degree of temporal and spatial regularity, each individual being characterized by a time-independent characteristic travel distance and a significant probability to return to a few highly frequented locations. They also show that the probability density function of individual travel distances is heavy tailed and also is different for different groups of users and similar inside each group. Furthermore, they plot the frequency of visiting different locations and show that it is well approximated by a power law. All these observations are in contrast with the random trajectories predicted by Levy flight and random walk models, and support the intuition behind SWIM. Also the authors of [16] are inspired by the work of Barabasi et al. They point out the following three rules of human mobility: 1) Nodes move more frequently and visit more locations if they have many friends; 2) users tend to visit a few locations where they spend the majority of their time; 3) users prefer shorter paths to longer ones. With these rules in mind, they propose HCMM, an improvement of their previous work in [10], [11]. They also include evaluation of temporal properties, in terms of intercontact times, of the traces generated by their model. In [24], the authors propose a mobility model that aims to reproduce real-world mobility traces, trying to capture group movements present in real-life mobility. The model is validated against real-world traces of vehicular networks, and the performance of the ADV and DSDV routing protocols is compared on both real and synthetic traces.

More recent works such as [25], [26] present other models for human mobility that are simple, and match well statistical properties of traces. However, these models have not been shown nor to generate community substructure such as those of real scenarios, neither to accurately validate protocols. Lastly, to the best of our knowledge no mobility model has been shown to have the capability to scale to larger scenarios in a consistent way.

3 Small World in Motion

The complexity of interpersonal relationships and the multitude of hobbies/interests that people have in a life that becomes more and more hectic make human mobility all but easy to model. In our vision, a model should be simple, easy to implement, and able to extrapolate key properties of human mobility. We cannot underestimate the importance of having a *simple* model. A simple model is easier to understand, can be useful to distill the fundamental ingredients of human mobility, can be easier to implement, easier to tune (just one or few parameters), and can be useful to support theoretical work. We are also looking for a model that generates traces with the same statistical properties that real traces have. Statistical distribution of intercontact time and number of contacts, among others, are useful to characterize the behavior of a mobile network. A model that generates traces with statistical properties that are far from those of real traces is probably useless. Simultaneously, a good model should also be able to generate similar social behavior among nodes to that of real life. However, it is important to keep in mind that matching statistical properties is not our final goal. It can even be misleading—if in the quest for matching a large number of statistical indicators we design a complicated model that is hard to use and understand, we are not doing a good job. It is much more important that the model is accurate in predicting the performance of network protocols on real networks. If a protocol performs well (or bad) in the model, it should also perform well (or bad) in the real network. As accurately as possible.

Lastly, we are looking for a model that, starting from a small (large) well-known scenario, can generate *large (small) scale* versions of it. A model that we can trust and use to assess the performance of forwarding protocols on networks whose size far exceeds (or is way below) the size of any available real experiment.

None of the mobility models in the literature meets *all* of these properties. The random waypoint mobility model is simple, but its traces do not look real. Some of the other protocols we reviewed in the related work section can indeed produce traces that have good statistical properties, at least with respect to some of the statistics, but are far from being simple. And, as far as we know, no model has been shown to predict real-world performance of community-based protocols accurately, and no model has been validated on larger scenarios (larger than known real traces) in a consistent way.

3.1 The Intuition

According to studies by the Temple University, Phi, USA,[1] the five topmost factors that impact peoples' choice when reallocating are safety, costs, good (high level) schools, convenience to shopping, proximity to work, proximity to family. While it is difficult to reinterpret safety and costs in terms of a mobility model where simplicity is the main requirement, the other factors suggest that people do consider proximity and popularity (high level of schools, good shopping, for example) when making decisions about mobility. People tradeoff these two basic elements in everyday mobility as well—the best supermarket/school or the most popular restaurant that are also not far from home, for example. It is unlikely (though not impossible) that we go to a location that is far from our place and that is not so popular, or interesting. Not only that, usually there are just a few places where a person spends a long period of time (for example, home and work office or school), whereas there are lots of places where she stays less, like, for example, post office, bank, cafeteria, and so on. So, supported by the studies in [12], [27], we expect that the wait-time follows a bounded power-law distribution. These are the two basic intuitions SWIM is built upon. Of course, tradeoffs humans face in their everyday life are usually much more complicated, and there are plenty of unknown factors that influence mobility. However, we will see that simple rules—tradingoff proximity and popularity, and distribution of waiting time—are enough to get a mobility model with a number of desirable properties and an excellent capability of predicting the performance of forwarding protocols. These simple rules, our model is based upon, are enough to make typical properties of real traces emerge, just naturally.

1. http://americashometown.blogspot.it/2005/12/why-people-choose-to-live-where-they.html.

3.2 The Model in Details

In SWIM, to each node is assigned a so-called *home*—a randomly and uniformly chosen point over the network area. The domain is continuous, so we divide the network area into many small contiguous squared cells that represent possible destinations. The size of the cells depends on the transmitting range r of the nodes—the cell diagonal equals r; this way, nodes that are in the same cell at the same time are able to communicate. Each node can, thus, easily build a map of the network area. That said, every node independently assigns to every destination cell a *weight* that grows with the popularity of the place and decreases with the distance from the node's home. The node chooses its destination cell randomly and proportionally with its weight. The exact destination point (remind that the network area is continuous) is taken uniformly at random over the cell's area.

More specifically, let A be one of the nodes and h_A its home. Let C be one of the possible destination cells. We denote with $seen(C)$ the number of nodes that node A encountered in C the last time it reached C. This number is 0 at the beginning of the simulation and it is updated each time node A reaches a destination in cell C. The weight that node A assigns to cell C is as follows:

$$w(C) = \alpha \cdot distance(h_A, C) + (1 - \alpha) \cdot seen(A, C). \qquad (1)$$

Informally, $seen(A, C)$ and $distance(h_A, C)$ measure, respectively, the popularity and the distance of cell C from the point of view of node A. Constant $\alpha \in [0; 1]$ tradeoffs distance from home and popularity. The larger α, the more a node will tend to go to places near its home and to meet neighbors. Conversely, the smaller α, the more a node will tend to go to "popular" places and to meet large "crowds of nodes." Of course, there is no "correct" scenario. Both are correct, they simply model different social structures.

Let h_A, x, and x_j be, respectively, node's A home-point, and the center of cells C and C_j. Let also r be the nodes' radius and d be the nodes' density in the network area (computed as a function of r and the total number of nodes). The *seen* and the *distance* functions of (1) are defined as follows:

$$seen(A,C) = \frac{1 + \frac{1}{d} TSeen(A,C)}{max_j\{1 + \frac{1}{d} TSeen(A,C_j)\}}, \quad (2)$$

where $TSeen(A,C)$ and $TSeen(A,C_j)$ denote the number of nodes A has encountered during all its visits, respectively, in C and C_j, and

$$distance(A,C) = \frac{\frac{1}{(1+\frac{1}{r}\|h_A - x\|)^2}}{max_j\left\{\frac{1}{(1+\frac{1}{r}\|h_A - x_j\|)^2}\right\}}. \quad (3)$$

As can be noticed from (2), node density plays a crucial role in a given cell's popularity. Indeed, a given density value has the same impact on popularity, regardless of network area. As well, the *seen* function that we propose depends on the total number of encounters a node has seen during all the visits in a cell. This tend to build a stable mobility pattern over time: After an initial setup period, nodes tend to belong to a static set of communities.

The *distance(A,C)* function (3) depends on the communication range r of the nodes. The model is built so that r determines the number of possible cells. Thus, it directly impacts the network area map for nodes. It is easy to see that the *distance* function of (3) *does* scale with the scaling of network area.

After a destination is chosen, a node moves toward it following a straight line and with a constant speed that is proportional to the distance between the starting point and the destination. In particular that means that nodes finish each leg of their movements in constant time. This can seem quite an oversimplification, however, it is useful and also not far from reality. Useful to simplify the model; not far from reality because we are used to move slowly (maybe walking) when the destination is nearby, faster when it is farther, and extremely fast (maybe by car) when the destination is far off. When reaching destination, the node decides how long to remain there by using a bounded (also known as truncated) power law. As discussed above, this is a key observation coming from real experiments.

3.3 Power Law and Exponential Decay Dichotomy

In a recent work [4], it is observed that the distribution of intercontact time in real-life experiments shows a so-called dichotomy: First a power law until a certain point in time, then an exponential cutoff. In [6], the authors suggest that the cutoff is due to the bounded domain where nodes move. In SWIM, intercontact time distribution shows exactly the same dichotomy. Our experiments show that, if the model is properly tuned, the distribution is strikingly similar to that of real-life experiments.

Here, we prove mathematically that the distribution of intercontact time of nodes in SWIM has an exponential tail (cut-off). Later, we will see experimentally that the same distribution has indeed a head distributed as a power law. Note that the proof has to cope with a difficulty due to the social nature of SWIM—every decision taken in SWIM by a node *does not* depend only on its own previous decisions, but also on other nodes' decisions. Where a node goes affects, where it will choose to go in the future, and where other nodes will choose to go in the future. So, SWIM has no renewal intervals and nodes never "forget" their past.

In the following, we will consider two nodes A and B. Let $A(t)$, $t \geq 0$, be the position of node A at time t. Similarly, $B(t)$ is the position of node B at time t. We assume that at time 0 the two nodes are leaving visibility after meeting. That is, $\|A(0) - B(0)\| = r$, $\|A(t) - B(t)\| < r$ for $t \in 0^-$, and $\|A(t) - B(t)\| > r$ for $t \in 0^+$. Here, $\|\cdot\|$ is the euclidean distance in the square. The intercontact time of nodes A and B is defined as $T_I = \inf_{t>0}\{t : \|A(t) - B(t)\| \leq r\}$.

Observation 1. For all nodes A and for all cells C, the distance function $distance(A,C)$ returns at least $\mu > 0$.

Theorem 1. *If* $\alpha > 0$, *the tail of the intercontact time distribution between nodes A and B in SWIM has an exponential decay.*

Proof. To prove the presence of the exponential cutoff, we will show that there exists constant $c > 0$ such that $\mathbb{P}\{T_I > t\} \leq e^{-ct}$, for all sufficiently large t. Let $t_i = i\lambda$, $i = 1, 2, \ldots,$ be a sequence of times. Constant λ is large enough that each node has to make a waypoint decision in the interval between t_i and t_{i+1} and that each node has enough time to finish a leg. This is possible because waiting time at waypoints is bounded above and nodes complete each leg of movement in constant time. The idea is to take snapshots of nodes A and B and see whether they see each other at each snapshot. However, in the following, we also need that at least one of the two nodes is not moving at each snapshot. So, let

$$\delta_i = \min\{\delta \geq 0$$
$$: \text{either } A \text{ or } B \text{ is at a waypoint at time } t_i + \delta\}.$$

Clearly, $t_i + \delta_i < t_{i+1}$, for all $i = 1, 2, \ldots$.

We take the sequence of snapshots $\{t_i + \delta_i\}_{i>0}$. Let $\varepsilon_i = \{\|A(t_i + \delta_i) - B(t_i + \delta_i)\| > r\}$ be the event that nodes A and B are not in visibility range at time $t_i + \delta_i$. We have that

$$\mathbb{P}\{T_I > t\} \leq \mathbb{P}\left\{\bigcap_{i=1}^{\lfloor t/\lambda \rfloor - 1} \varepsilon_i\right\} = \prod_{i=1}^{\lfloor t/\lambda \rfloor - 1} \mathbb{P}\{\varepsilon_i | \varepsilon_{i-1} \ldots \varepsilon_1\}.$$

Consider $\mathbb{P}\{\epsilon_i | \epsilon_{i-1} \ldots \epsilon_1\}$. At time $t_i + \delta_i$, at least one of the two nodes is at a waypoint, by definition of δ_i. Say node A, without loss of generality. Assume that node B is in cell C (either moving or at a waypoint). During its last waypoint decision, node A has chosen cell C as its next waypoint with probability at least $\alpha\mu > 0$, thanks to Observation 1. If this is the case, the two nodes A and B are now in visibility. Note that the decision has been made after the previous snapshot, and that it is not independent of previous decisions taken by node A, and it is not even independent of previous decisions taken by node B (since the social nature of decisions in SWIM). Nonetheless, with probability at least $\alpha\mu$ the two nodes are now in visibility. Therefore,

$$\mathbb{P}\{\varepsilon_i | \varepsilon_{i-1} \ldots \varepsilon_1\} \leq 1 - \alpha\mu.$$

So,

$$\mathbb{P}\{T_I > t\} \leq \mathbb{P}\left\{\bigcap_{i=1}^{\lfloor t/\lambda \rfloor - 1} \varepsilon_i\right\} = \prod_{i=1}^{\lfloor t/\lambda \rfloor - 1} \mathbb{P}\{\varepsilon_i | \varepsilon_{i-1} \ldots \varepsilon_1\}$$
$$\leq (1 - \alpha\mu)^{\lfloor t/\lambda \rfloor - 1} \sim e^{-ct},$$

for sufficiently large t. □

3.4 The Simulation Environment

To evaluate SWIM, we built a discrete event simulator of the model (see the website for SWIM [28]). The simulator takes as input:

- n: the number of nodes in the network;
- r: the transmitting radius of the nodes;
- the simulation time in seconds;
- coefficient α that appears in (1); and
- the distribution of the waiting time at destination.

The output of the simulator is a text file containing records on each main event occurrence. The main events of the system and the related outputs are:

- *Meet* event. When two nodes are in range with each other. The output line contains the ids of the two nodes involved and the time of occurrence.
- *Depart* event. When two nodes that were in range of each other are not anymore. The output line contains the ids of the two nodes involved and the time of occurrence.
- *Start* event. When a node leaves its current location and starts moving toward destination. The output line contains the id of the location, the id of the node, and the time of occurrence.
- *Finish* event. When a node reaches its destination. The output line contains the id of the destination, the id of the node, and the time of occurrence.

During the simulation each node A keeps a vector $TSeen(A, C)$ updated. So, in every moment A is able to compute the value $seen(A, C_i)$ for all the cells C_i. Note that the nodes do not necessarily agree on what is the popularity of each cell. Indeed, usually they do not because nodes visit cells at different times. As mentioned earlier, it is not necessary to keep in memory the whole vector, without changing the qualitative behavior of the mobile system. However, the four real scenarios we will consider next are not large enough to cause any real memory problem. Vector $TSeen(A, C)$ is updated at each *Finish* and *Start* event, and is not changed during movements.

Lastly, note that people do not tradeoff proximity for popularity in the same way. Take a salesman, for example—he moves frequently from one town to another, or from one building to another. Surely, he has a different mobility pattern compared to a high-school teacher that tends to move in a more repetitive way. SWIM is able to simulate these scenarios too, simply by setting different α values to different nodes. Nonetheless, the scenarios we simulate in this work involve only people with similar jobs/interests (students or conference attendees), so here we set α to be the same for all nodes.

3.5 Generating Large Scenarios with SWIM

Obtaining large and trustworthy synthetic mobility traces is both important and challenging. It is important to assess networking protocols on data sets larger than those available today and, thus, check their scalability; it is challenging because it is not clear how a large mobility trace should look like by looking just at the few available and small real-world data sets. Here, we propose a methodology.

To generate mobility traces with SWIM, we choose the parameters and let the model generate traces as long as we need. In the literature, it is customary to choose the parameters in such a way that the mobility pattern is similar, in some precise statistical sense, to a real data set. For example, the data set collected during the Infocom conference in 2006 (in the experimental section of this paper, we show how to do it for this real data set among others). In this way, we can build a model that looks like Infocom 2006 with $n = 78$ nodes and density ρ (tuned with a large set of experiments). This is already a very useful thing to do, we are now able to generate traces that are much longer than the three days of the conference in a sound way.

Here, we consider the problem of generating traces for the same scenario in which the number of nodes is $N > n$. If the basic assumption of SWIM is correct (people tradeoff popularity of places and vicinity with a parameter α), it is enough to replace the number of nodes n in the original model with N. We can also assume that transmission range does not change with the number of the nodes. The only issue, which is not obvious indeed, is how density $\rho(N)$ changes as N grows and, consequently, how the area of the network changes as N grows.

Actually, it is impossible to give an answer to this problem. It is like predicting the future growth of a mobile community. Our effort in this direction is to build a model that is able, in a simple way, to generate scaled versions of nowadays networks. It seems reasonable to bound the possible future of a growing community by using two extremes that we define in this paper: The Phoenix model and the Manhattan model. In the *Phoenix model*, $\rho_P(N) = \rho$ for all $N > n$ (recall that ρ is the density that has experimentally been shown to be appropriate for the scenario when the number of nodes is n). Speaking in metaphorical terms, this is the case when a town grows in size without creating denser agglomerates and just covering a larger geographical area. In the *Manhattan model*, $\rho_M(N) = N\rho/n$ for all $N > n$. In this model, as the network grows more people populate the same geographical area. The place is just much more crowded, and that means that every node meet many more other nodes in the same unit of time and that people mix more (it is more common to meet people that are not in your circle of friends).

When assessing the performance of networking protocols, a fundamental property to check is scalability. This is one of the contributions of this paper, showing how some of the protocols that are the state of the art perform on large networks—larger than any real data. Thanks to SWIM, we are able to show the performance under the Phoenix and the Manhattan models. If a protocol shows good performance in larger and larger networks under both models, then we can have some confidence that the model has good scalability. We still do not know how large mobile networks are going to be. However, we can predict the scalability of the protocols under the various hypotheses that are represented by these models. This allows us to study, for the protocols we have now, the performance in reasonable future scenarios. Clearly, a more comprehensive experiment can consider a class of density functions ρ such that

TABLE 1
The Three Experimental Data Sets

Dataset	Camb.	Inf. 05	Inf. 06	Dart.
Device	iMote	iMote	iMote	SPh,laptops
Network type	Bluetooth	Bluetooth	Bluetooth	AP
Duration (days)	11	3	3	60
Devices number	36	41	78	1146

$\rho_P(N) \leq \rho(N) \leq \rho_M(N)$ and, thus, understand under what conditions of scalability the protocol has good performance.

4 Experimental Results

To show the accuracy of SWIM in simulating real-life scenarios, we will compare SWIM with four traces gathered during experiments done with real devices carried by people. We will refer to the real traces as *Cambridge*, *Infocom 05*, *Infocom 06*, and *Dartmouth*. Characteristics of these data sets such as intercontact times and contact distribution have been observed in several previous works [2], [29], [3]:

- In *Cambridge* [8], [30], the authors used Intel iMotes to collect the data. The iMotes were distributed to two groups of students (Year1 and Year2) of the University of Cambridge and were programmed to log contacts of all visible mobile devices. Also, a number of stationary nodes were deployed in various locations around the city of Cambridge United Kingdom. The data of the stationary iMotes will not be used in this paper. The number of mobile devices used is 36 (plus 18 stationary devices). This data set covers 11 days.
- In *Infocom 05* [8], [31], the same devices as in *Cambridge* were distributed to students attending the Infocom 2005 student workshop. Participants belong to different social communities (depending on their country of origin, research topic, etc.) The number of devices is 41. This experiment covers approximately three days.
- In *Infocom 06* [8], [31], the scenario was similar to Infocom 05 except that the scale is larger, with 78 participants. Participants were selected so that 34 out of 78 form four subgroups by academic affiliation: ParisA with 10 participants, ParisB with four participants, Lausanne five participants, and Barcelona 15 participants. In addition, 20 long range iMotes were deployed at several places in the conference site to act as access points. However, the data from these fixed nodes are not used in this paper.
- *Dartmouth* [32] includes SNMP logs from the access points (smartphones and laptops) across the Dartmouth College campus from April 2001 to June 2004. To generate user-to-user contacts from the data set, we follow the popular consideration in the literature that devices associated with the same AP at the same time are assumed to be in contact [3]. We consider activities from the 5th of January to the 6th of March 2004, corresponding to a 2-month period during which the academic campus life is reasonably consistent.

Further details on the real traces are shown in Table 1.

TABLE 2
Tuning Parameters

Scenario	Camb.	Inf. 05	Inf. 06	Dart.
Radius	.05	.04	.04	.013
Duration (days)	11	3	3	60
Number of devices	36	41	78	1146
Value of α	.8	.7	.7	.6
Waiting time slope	1.35	1.35	1.35	1.65
Waiting time bound	24h	12h	12h	277h

4.1 Tuning SWIM

Each parameter in SWIM has an impact on the outcome of the simulation. Table 2 shows, in details, the parameters we have used to tune SWIM when simulating each of the real scenarios considered. Here, we explain the tuning methodology we used.

4.1.1 Number of Nodes, Area, and Radius

First, the SWIM simulation area is fixed, 1×1. Parameters such as the number of nodes and the node radius are directly taken from the real setting. If, for example, the contacts are bluetooth (WiFi) based, the radius is set to emulate bluetooth (WiFi) communication range. Then, it is scaled according to the experimental area. Finally, the simulation area is divided into contiguous and equally sized cells, whose diagonal equals the radius.

In Infocom 06, for example, the number of nodes is set to 78. We then set the radius to 0.04, as an approximate proportion between bluetooth range and an estimation of the conference area (a hotel of around 700 m^2). In the case of Dartmouth, being a campus (surface around 2 km^2), we set the radius to 0.013. Clearly, this automatic setting might not yield the best results, as radius influences the number of contacts. In particular, larger radius means higher number of "random" contacts—contacts that happen when the nodes move from one point of interest to another—and vice versa. However, we have observed that these parameters follow intuition very precisely, and that this way of setting simulation area, nodes, and communication radius yields very accurate results. Indeed, small differences in these parameters create small differences in the traces, thus allowing to tune the model systematically.

4.1.2 Waiting Time Distribution

The parameters of the waiting time come directly from the traces in an automatic way. In Cambridge, Infocom 05 and Infocom 06 the head distribution of intercontact times has a slope of 1.35; whereas in Dartmouth, it is 1.65. Accordingly, we set the slope of the wait time distribution to be exactly the one observed from the real trace. Similarly for the cutoff—it is set to match the length of the power-law head of the intercontact time distribution of the respective real scenario. The values are: 24 h for Cambridge, 12 h for both Infocom scenarios, and, 11.5 days (277 h) for the Dartmouth scenario.

4.1.3 Parameter α: Local Small Restaurant or VIP Bar?

To understand how parameter α influences the results we setup the following experiment: We simulate a 100 node network by keeping all parameters fixed but α, which is

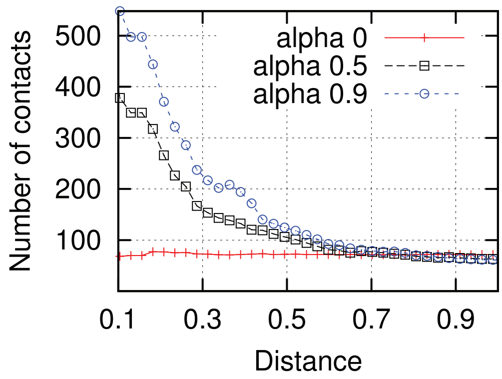


Fig. 1. How does the number of contacts with other nodes depend on the mutual home-point distance for various values of α.

set to three different values: 0, 0.5, and, 0.9. Then, we compute the number of contacts of a randomly chosen node A with other nodes in the network as a function of the distance of their homes. We plot the results in Fig. 1. The result confirms our intuition: The bigger α, the more frequently nodes meet their neighbors. This is due to the fact that for high values of α nodes tend to restrict their movement to cells nearby their home. With lower α, the phenomenon is attenuated and for $\alpha = 0$ the meeting rate does not depend at all on the distance of the homes (from Fig. 1 the trend of the meeting probability when this distance varies is almost uniform).

Playing with the α parameter, and only this one, it is possible to boost social aspects of a well-known test scenario, or to boost geographical aspects of the same scenario. In Cambridge, the nodes are freshmen and sophomores at the Cambridge University. It is reasonable to think that freshmen meet freshmen and that sophomores meet sophomores more frequently. Indeed, we have seen that $\alpha = 0.8$ (giving more weight to geographical aspects) yields traces with statistical properties very close to the real one. In conferences, participants typically meet people that share affiliation or research interests. However, there are occasions that favor social mixing (e.g., social events, coffee brakes, etc.). This is why, experimentally, a smaller $\alpha = 0.7$ proved to work best, and yield a synthetic trace that better matches the statistical properties of the real one. Dartmouth is different—the AP-based contacts make so that people that go to the same place meet even though they might not share much. Nonetheless, students with same interests (e.g., taking the same classes) still tend to meet more often between them than with other students. In this case, which has higher mixing, the best α has been 0.6. Clearly, setting parameter α is not an automatic process. It is, thus, important to observe that in SWIM the results are always consistent with intuition, and that the number of parameters that have to be set in a nonautomatic way is very limited.

4.2 SWIM versus Reality: Statistical Properties

Here, we present experimental results comparing statistical properties of the real scenarios with respect to SWIM. The parameters are shown in Table 2. We will call the four synthetic versions of Cambridge, Infocom 05, Infocom 06, and Dartmouth, respectively, SWIM 36, SWIM 41, SWIM 78, and SWIM 1146, where the number refers to the number of nodes in the scenario. It is particularly interesting that we might as well have got the (almost) exact parameters for SWIM 78 (the synthetic version of Infocom 06) by scaling SWIM 41 (the synthetic version of Infocom 05) according to the Manhattan model (constant area, higher density). Indeed, we can conjecture that the two real scenarios run in an area of approximately the same size, with roughly double density because the number of devices distributed is roughly the double. This simple fact is a good support to our methodology.

For each of the experiments, we consider the following metrics: Intercontact time CCD function, contact distribution per pair of nodes, and number of contacts per pair of nodes. The intercontact time distribution is important in mobile networking because it characterizes the frequency with which information can be transferred between nodes. It has been studied for real traces in a large number of previous papers [2], [3], [29], [6], [4], [10], [33]. The distributions of contact durations and contact frequency per node-pairs are also important. Indeed, they represent a way to measure relationship between people. As also discussed in [34], [35], [8], it is natural to think that if two people spend time together and meet frequently then they are familiar to each other. Familiarity is important in detecting communities, which may help improve significantly the design and performance of forwarding protocols in mobile environments [8].

In Fig. 2, we show the results for Cambridge and for SWIM 36 (the synthetic version of Cambridge). Moreover, we have considered SWIM-M 360 that is a larger version of

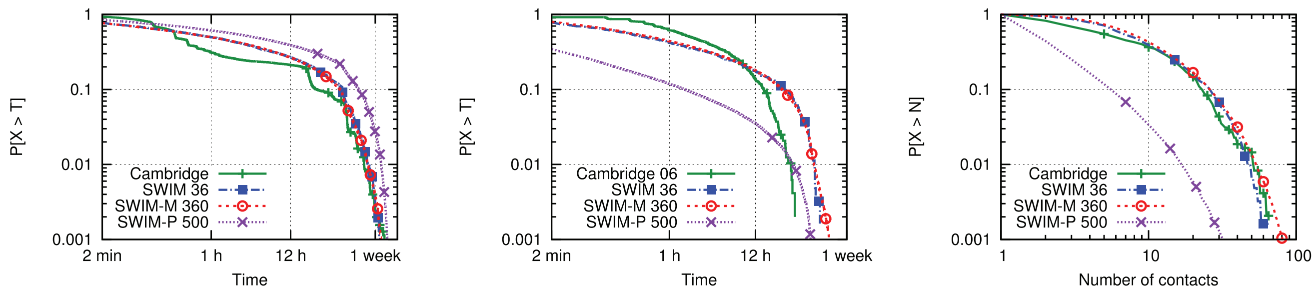
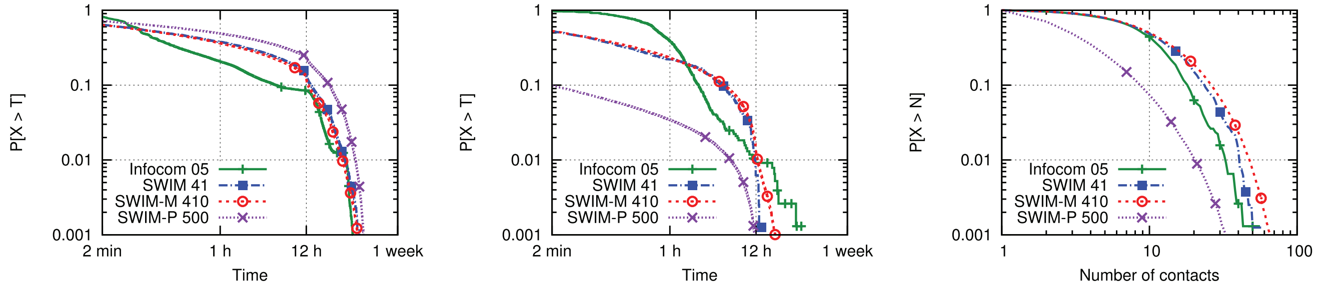
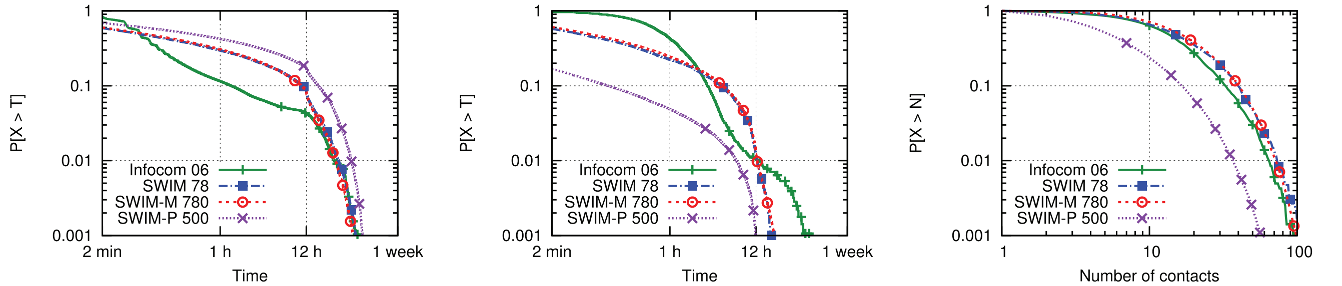


Fig. 2. SWIM and Cambridge. Cambridge is the real scenario, SWIM 36 is the synthetic version of Cambridge, SWIM-P 500 is Cambridge with 500 nodes according to the Phoenix model, and SWIM-M 360 is Cambridge with 360 nodes according to the Manhattan model.



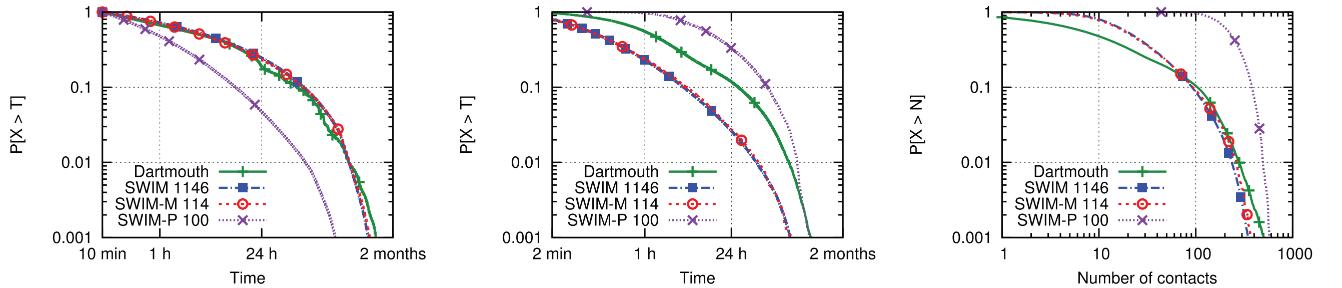
(a) Inter-contacts (b) Contact duration per node pairs (c) Contact number per node pairs

Fig. 3. SWIM and Infocom 05. Infocom 05 is the real scenario, SWIM 41 is the synthetic version of Infocom 05, SWIM-P 500 is Infocom 05 with 500 nodes according to the Phoenix model, and SWIM-M 410 is Infocom 05 with 410 nodes according to the Manhattan model.



(a) Inter-contacts (b) Contact duration per node pairs (c) Contact number per node pairs

Fig. 4. SWIM and Infocom 06. Infocom 06 is the real scenario, SWIM 78 is the synthetic version of Infocom 06, SWIM-P 500 is Infocom 06 with 500 nodes according to the Phoenix model, and SWIM-M 780 is Infocom 06 with 780 nodes according to the Manhattan model.



(a) Inter-contacts (b) Contact duration per node pairs (c) Contact number per node pairs

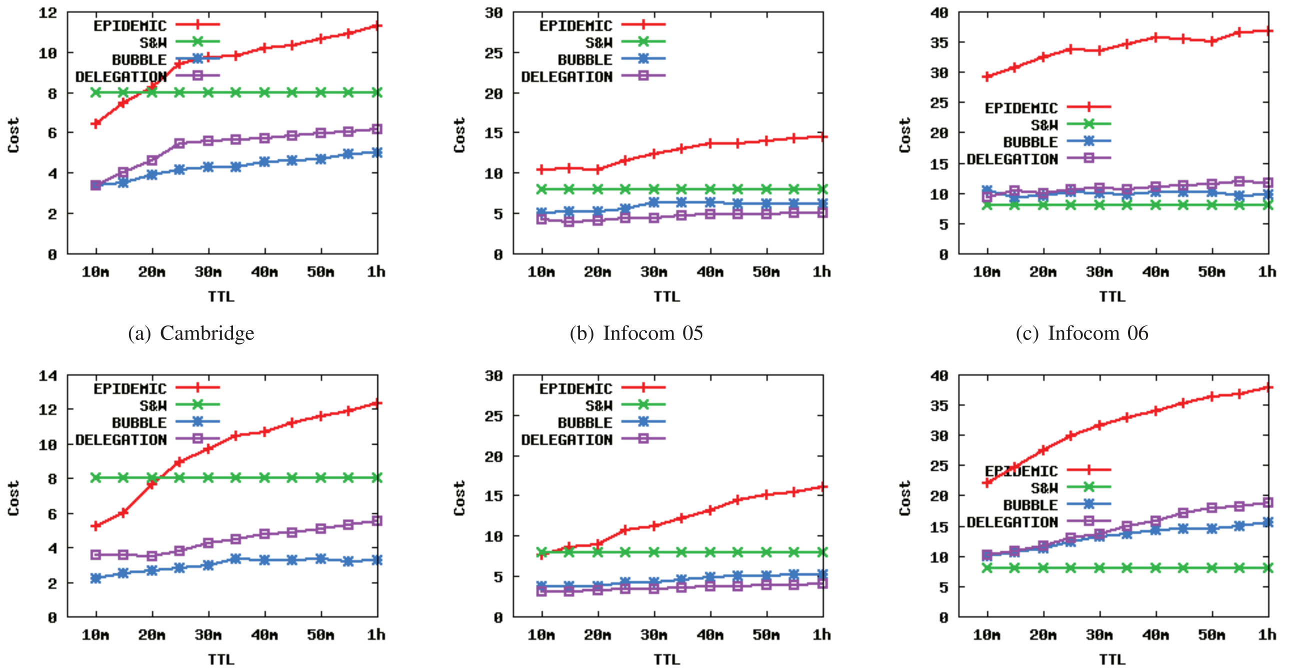
Fig. 5. SWIM and Dartmouth. Dartmouth is the real scenario, SWIM-1146 is the synthetic version of Dartmouth, SWIM-P 100 is Dartmouth with 100 nodes according to the Phoenix model, and SWIM-M 114 is Dartmouth with 114 nodes according to the Manhattan model.

Cambridge with 10 times the number of nodes according to the Manhattan model, and SWIM-P 500 a version with 500 nodes according to the Phoenix model. Similarly, Figs. 3 and 4 show the results for Infocom 05, Infocom 06, their synthetic versions, and the larger scenarios built according to the Manhattan and the Phoenix models. In Fig. 5, we show the results for Dartmouth. As the figures suggest, SWIM yields synthetic traces with statistical properties that are similar to the real ones. To strengthen this claim we show, in Table 3, the Jensen-Shannon divergence [36] between a given distribution in one of the real scenarios and its synthetic alter ego, for all the distributions considered. The Jensen-Shannon divergence measures the similarity of two probability distributions and takes values in $[0;1]$, higher values mean higher divergence. We note that all values are low, which confirm what we observed from the figures. Note that the same choice of parameters gets good results for all the metrics under consideration at the same time.

TABLE 3
Jensen-Shannon Divergence between Distributions of the Real and SWIM Traces

Trace	Inter-contacts	Contacts Duration	Contact Number
Cambridge	.058	0.15	0.004
Infocom 05	.062	0.21	0.005
Infocom 06	.049	0.18	0.0114
Dartmouth	.028	0.11	0.073

In the figures, we also show the behavior of the Phoenix (constant density) and Manhattan (constant area) models. Let us first discuss the Phoenix model: If we consider two *arbitrary* nodes, it is more likely that they meet less frequently as the number of nodes grows (and so the area). As a consequence, the intercontact time should decay slower, while the contact-duration and the number of contacts should decay faster. Intuition is fully confirmed by the experimental results (see Figs. 2a, 2b, 2c, 3a, 3b, 3c, and 4a for the intercontact times distribution and Figs. 2b, 2c, 3a,



(a) Cambridge (b) Infocom 05 (c) Infocom 06

(d) SWIM-Cambridge (e) SWIM-Infocom 05 (f) SWIM-Infocom 06

Fig. 6. Average cost of forwarding protocols. Cambridge and Infocom 05 (06) are the results on the real traces. SWIM-Cambridge and SWIM-Infocom 05 (06) are the results on the simulated traces.

3b, 3c, 4a, and 4b, and Figs. 2c, 3a, 3b, 3c, 4a, 4b, and 4c for the contact duration and the contact-number distributions). In the same figures, we can see that the Manhattan model is different. Since the area is the same when the number of nodes grows, the distribution of intercontact time, contact duration, and number of contacts between any arbitrary pair of nodes should not change. It is just a more crowded world. This is also completely supported by our results.

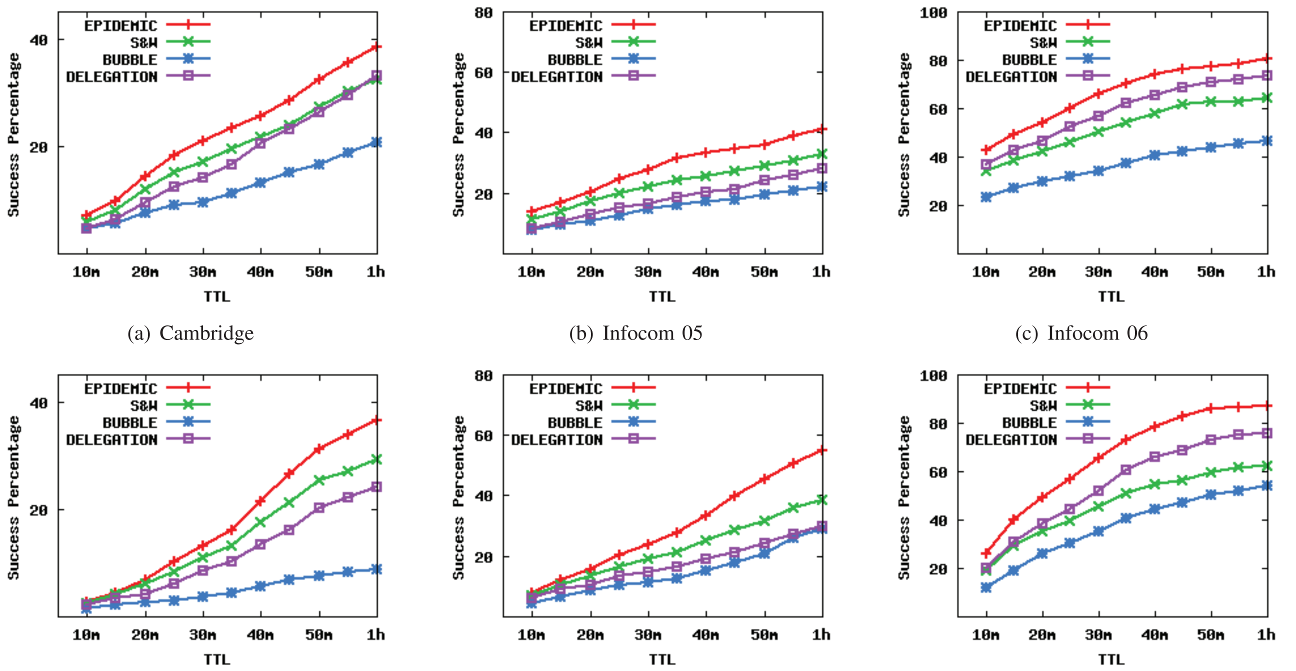
For the Dartmouth case, we downscale: We scale to obtain smaller networks—this trace is large enough (1,146 nodes) to make it possible. For the Manhattan case, we keep the area constant and lower the nodes number (so to lower the density), whereas, for the Phoenix case we lower the number of nodes yet keeping the density constant (so we lower the area). As in the up-scaling case (getting enlarged traces), from the graphics we observe that, for the Manhattan scaling (lower density, constant area), the distributions are preserved. Whereas for the Phoenix scaling (constant density, smaller area), the effect in the distributions is exactly the opposite of that of the up-scaling: The smaller area makes so that nodes couple meet more frequently, and for longer times if averaged with all the nodes in the network. So, intercontact times decay faster, while the contact duration and the number of contacts decay slower (see Figs. 5a, 5b, and 5c for, respectively, the intercontact times, contact duration, and the contact-number distributions).

4.3 Protocol Performance

Now, we get to a fundamental aspect for every model. We want to show that SWIM is good to predict the performance of forwarding protocols. We describe the experimental results of SWIM and four forwarding protocols for DTNs: Epidemic Forwarding [20], Delegation Forwarding [19], Spray&Wait [21], and BUBBLE [8]. In the experiments, we use exactly the same tuning used in the previous section. That is, the parameters input to SWIM are not "optimized" for each of the forwarding protocols, they are just the same that has been used to fit real traces with synthetic traces.

For the evaluation, we use the same assumptions and the same way of generating traffic to be routed as in [19]. For each trace and forwarding protocol, a set of messages is generated with sources and destinations chosen uniformly at random, and generation times form a Poisson process averaging one message every 4 seconds. The nodes are assumed to have infinite buffers and carry all message replicas they receive until the end of the simulation—this is in accordance with the literature on these protocols. The comparison is done in terms of *success percentage* (rate of messages delivered to destination) and *cost* (average number of replicas per delivered message) as a function of message TTL (time to leave). Message traffic follows a uniform traffic pattern (source-destination distributed uniformly at random among network nodes). As in [19], we isolated 3-hour periods for each data trace (real and synthetic) for our study. Each simulation runs, therefore, 3 hours. To avoid end-effects no messages were generated in the last hour of each trace.

Figs. 6 and 7 show how the forwarding protocols perform in both real and synthetic traces, generated with SWIM. The first observation that we make is that the trend of the protocols in the real scenarios is the same with that of the respective synthetic ones—the ones that perform better in the real world do so also in the SWIM-generated one. This support the claim that SWIM is an excellent model for protocol validation. In particular, this is also true for complex forwarding protocols such as BUBBLE that depend on the structure of the network in social communities.



(a) Cambridge (b) Infocom 05 (c) Infocom 06

(d) SWIM-Cambridge (e) SWIM-Infocom 05 (f) SWIM-Infocom 06

Fig. 7. Average success percentage of forwarding protocols. Cambridge, Infocom 05 (06) are the results on the real-traces. SWIM-Cambridge, and SWIM-Infocom 05 (06) are the results on the simulated traces.

Another important observation is that each protocol performs very similarly when validated both in SWIM and in the respective real trace. This is also confirmed by the results of Table 4, where we show the average error percentage of each protocol in SWIM compared to its performance on the respective real trace, for the Infocom 06 scenario. Note that the difference is small in both terms of cost and success percentage. (Unfortunately, due to space restrictions we omit the results for the other data sets; however, we stress that the trend is very similar.) So, we conclude that the performance of *all* protocols in the small-scale scenarios can be accurately predicted by running the protocols on the synthetic traces. Most importantly, this is not due to a customized tuning that has been optimized for these forwarding protocols, it is just the same output that SWIM has generated with the tuning of the previous section. This can be important methodologically: To tune SWIM on a particular scenario, you can concentrate on a few well known, important statistical properties like intercontact time, contact number, and duration. Then, you can have a good confidence that the model is properly tuned and usable to get meaningful performance estimation of a forwarding protocol.

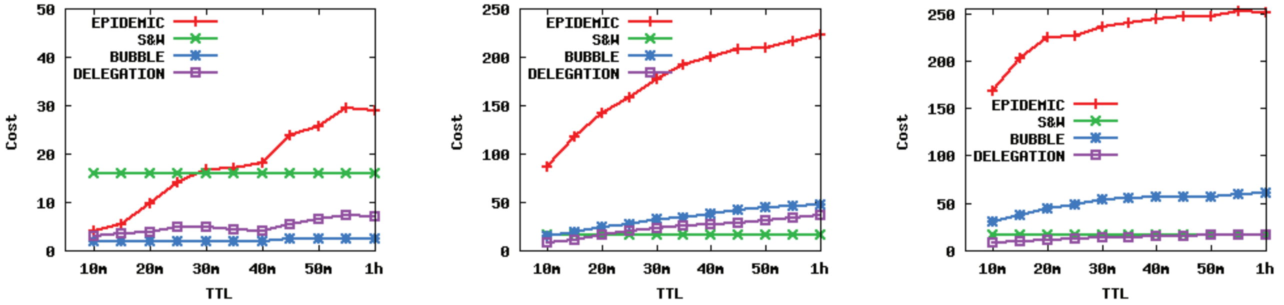
TABLE 4
Error Percentage of Protocol Performance (SWIM versus Infocom 06)

Protocol	Cost	Success
Epidemic forwarding	.08	.10
Delegation forwarding	.24	.11
BUBBLE Rap	.24	.15
Spray & Wait	0	0.12

Finally, to compare SWIM to the well-known RWP model, we setup the following experiment: we simulate with RWP one of the real scenarios considered in the paper—mainly, the Infocom 06 scenario—and we run on the RWP trace and on the SWIM trace Delegation Forwarding and Random Forwarding (when A and B meet, B is decided to be a relay of a message depending on the result of a coin toss). Whereas Delegation performs highly better than Random forwarding on the SWIM trace, there is no distinction between the performances of the two protocols on the RWP trace. This is because in RWP the nodes' movement is memoryless, and it does not follow any social rule. So, the fact that a given node has seen the destination soon or not does not give any information on what will happen in the future. This is why using Delegation, a social-based forwarding strategy, rather than a random strategy to forward messages does not make any difference. Conversely, in SWIM the movement is social-based-nodes tend to regularly go to cells nearby their home-points, and where they have met in the past many other nodes. Thus, social-based strategies (such as Delegation, in this case) perform particularly better with respect to random strategies. Again, we do not show the relative plots because of space constraints.

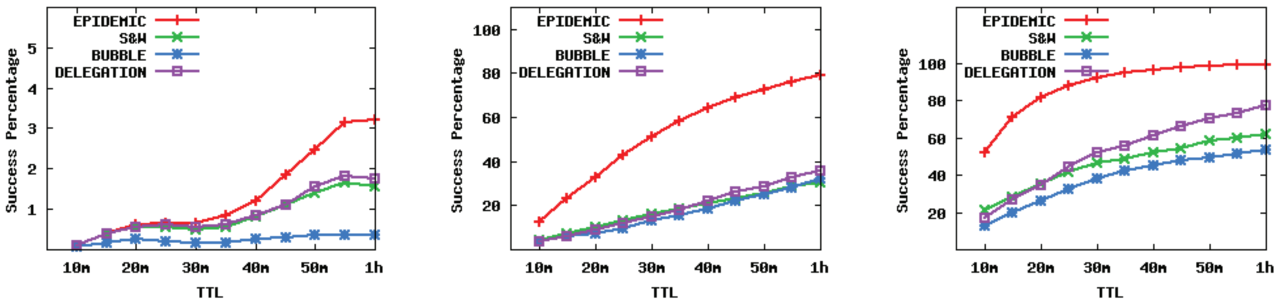
5 Scaling Capabilities of Forwarding Protocols

When designing a networking protocol, scalability is a most desired property. SWIM can be used to address this important question: How do well-known forwarding protocols perform in large-scale social mobile networks? To give an accurate answer to this question, we validate the previously considered forwarding protocols on large-scale



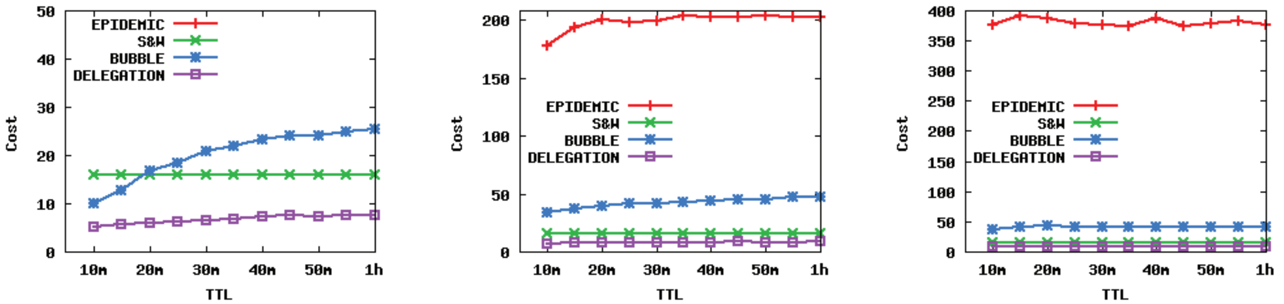
(a) Cambridge with SWIM-P (b) Infocom 05 with SWIM-P (c) Infocom 06 with SWIM-P

Fig. 8. Average cost of forwarding protocols on enlarged Phoenix scenarios (constant density, larger area).



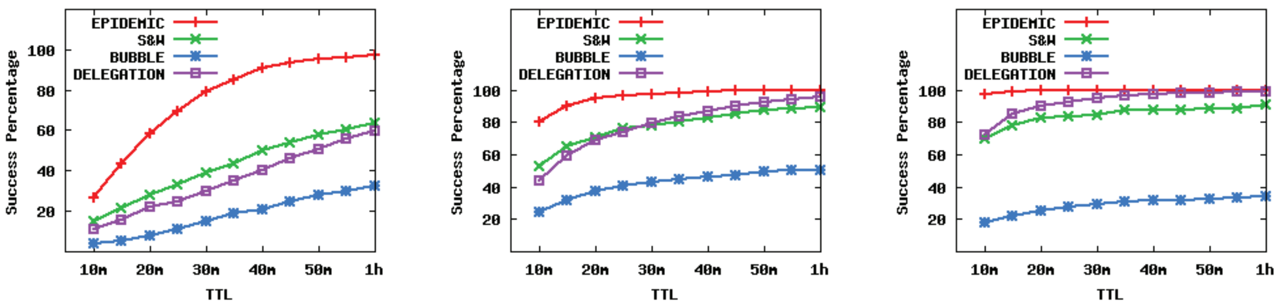
(a) Cambridge with SWIM-P (b) Infocom 05 with SWIM-P (c) Infocom 06 with SWIM-P

Fig. 9. Average success percentage of forwarding protocols on enlarged Phoenix scenarios (constant density, larger area).



(a) Cambridge with SWIM-M (b) Infocom 05 with SWIM-M (c) Infocom 06 with SWIM-M

Fig. 10. Average cost of forwarding protocols on enlarged Manhattan scenarios (higher density, constant area).

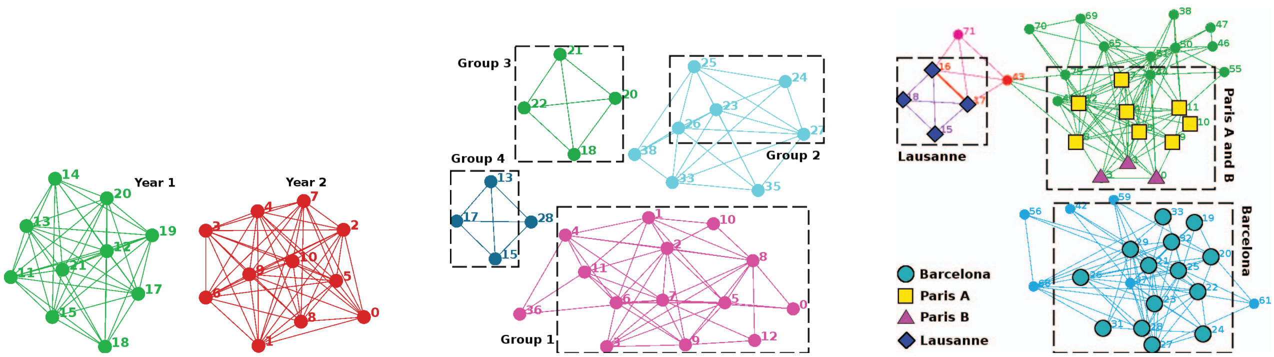


(a) Cambridge with SWIM-M (b) Infocom 05 with SWIM-M (c) Infocom 06 with SWIM-M

Fig. 11. Average success percentage of forwarding protocols on enlarged Manhattan scenarios (higher density, constant area).

SWIM-generated traces. The experimental setting is the same of the last section, whereas the Spray&Wait's limit on message copies differs from scenario to scenario, and is set following the suggestions of the authors in [21]. Again, we study the *success percentage* and *cost* for various TTL (time to leave). The results are presented in Figs. 8, 9, 10, and 11. Here, are our observations:

Scaling with the Phoenix model. When the number of nodes grows, the cost in terms of number of replicas is much higher, whereas the delivery rate drops considerably (compare Fig. 6 with Fig. 8 for the cost and Fig. 7 with Fig. 9 for the delivery rate). This is because when the network is enlarged by keeping the density constant, more hops are required to deliver a message (increasing the cost),



(a) Simulated Cambridge. (b) Simulated Infocom 05. (c) Simulated Infocom 06.

Fig. 12. Communities detected in the synthetic traces.

and simultaneously, the network area is much larger, which makes it more difficult to get a message to destination.

Scaling with the Manhattan model. The cost again is much higher for all protocols but so is the delivery ratio (compare Fig. 6 with Fig. 10 for the cost and Fig. 7 with Fig. 11 for the delivery rate). This scaling method yields much denser networks, so the many contacts help all protocols to deliver messages quickly. Nonetheless, this also makes more probable that a high number of replicas are generated in the network, so, the cost is increased.

It is worthy to notice that these effects are attenuated for BUBBLE, Delegation, and Spray&Wait, which adopt more sophisticated rules to keep the cost reasonably low. Also, Delegation Forwarding and Spray&Wait seem to offer the best tradeoff. They are not always the best when the network is small, but they show a good behavior when the network size grows compared both to Epidemic and to BUBBLE.

Overall, the experiments show that the quest for a scalable forwarding protocol for pocket switched network is still largely an open issue. Most probably, the techniques used in these protocols are excellent tools that can be used for larger and larger networks as well, but it seems that some new additional idea is needed to keep cost in terms of messages low enough and success rate reasonably high.

6 Ad Hoc Communities with SWIM

Many works have studied the communities that appear in traces of social mobile networks obtained from real experiments. To detect community substructures, the k-clique algorithm is widely used [37], [2], [3], [7]. The algorithm determines as belonging to the same community a union of adjacent cliques of k nodes sharing $k-1$ nodes [37]. In particular, this algorithm has been used in two of the scenarios we consider in this work—Infocom 06 and Cambridge [7]. The authors, which are also the ones who set up the experiments, have gathered information on the social relations of the participants. After detecting communities from the traces, they observe that the social relationships in real life have a good match with the ones uncovered from the traces by the k-clique algorithm.

In the Cambridge scenario, they detect two main communities of 11 members each that correspond to the students of the first and the second year. In the Infocom 06 scenario, they observe that most of the participants with the same academic affiliation (ParisA, ParisB, Lausanne, and Barcelona) do belong to the same community detected by the k-clique algorithm. Unlike the first two traces, Infocom 05 only contains partial information on the participants: There are four groups of, respectively, 10, six, four, four members each. It is not known how node IDs are mapped to participants, thus, which node is member of which group.

The next step of our study is to SWIM-generate these scenarios in an ad hoc manner, such that a given desired social structure is observed at the end of the simulation. Let us start with Cambridge 05. There are 36 students involved, grouped by academic year in two groups: Year1 and Year2. As we mentioned, in the real trace there are two clear communities of 11 students. To each community we assign a "center point" in the network area: $p_1 = (0.05; 0.05)$ and $p_2 = (0.95; 0.95)$ (for, respectively, groups Year1 and Year2). The members of each group is given a home-point obtained by perturbing the center point of their community with a Gaussian distribution of standard deviation of 0.01. The remaining 14 nodes are assigned a home-point obtained with a uniform distribution over the network area.

In Infocom 06, there are four communities (ParisA, ParisB, Lausanne, and Barcelona) of, respectively, 10, four, five, and, 15 members each. Therefore, to simulate this scenario we divide 34 nodes in four groups of as much members as in the real case. For each group, we assign a central point as follows: $p_1 = (0.01; 0.01)$ for ParisA, $p_2 = (0.013; 0.013)$ for ParisB, $p_3 = (0.95; 0.01)$ for Lausanne, and, $p_4 = (0.5; 0.95)$ for Barcelona. Note that the members of the two Paris groups are initially placed close, to simulate social connection among them. The members of each group is given a home-point obtained by perturbing the respective center point with a Gaussian distribution of standard deviation of 0.01. The remaining nodes are assigned home-points chosen uniformly and randomly over the network area.

Unlike the Cambridge and the Infocom 06 scenario, in Infocom 05 we have no exact information on the social relationships among participants. We have however information on the initial affiliation of some of the members (given by the authors of the experiment). So, in this case, we obtain the community information from the trace itself. We first run the k-clique algorithm on the Infocom 05 trace. The communities that we detect are consistent with the information we have on the experiment—four communities of, respectively, 10, 6, four, and four members each. This is not surprising. The k-clique algorithm is indeed one of the most used in the area to uncover social substructures from real traces that reflect well the social relationships in real life. Then, to simulate this scenario we feed the simulator with the information extrapolated from the k-clique communities uncovered in the real trace. We divide our nodes in four groups of as much members as in the real case. For each group, we assign a central point as follows: $p_1 = (0.95; 0.95)$ for group 1, $p_2 = (95; 0.05)$ for group 2, $p_3 = (0.05; 0.95)$ for group 3, $p_4 = (0.05; 0.05)$ for group 4. The members of each group is given a home-point obtained by perturbing the respective center point with a Gaussian distribution of standard deviation of 0.01. The remaining nodes are assigned home-points chosen uniformly and randomly over the network area.

The rest of the simulation parameters are set as described in Table 2. In particular, the choice of α is done based on the grade of relationship people have in the scenarios (conferences versus university): 0.8 and 0.7 for Cambridge and the two Infocom scenarios, respectively. Also, the choice of the waiting time bound is done based on the real traces intercontact time distribution's head. In the Cambridge case, it follows a power law for 24 hours, whereas in both Infocom scenarios for 12 hours.

In Fig. 12, we show the communities detected from the synthetic traces. As can be seen, in each simulated scenario the structure in communities reflects very well the real scenario: Nodes whose affiliation was emulated by assigning adjacent home-points result being members of the same community detected after the simulation. This means that SWIM preserves initial "social relationships" among nodes in the same way as a real social mobile network does and that it can be used to recreate traces with known community structures.

7 Conclusions

In this paper, we have presented SWIM, a mobility model that we can use to generate small mobile worlds. SWIM is very simple and it generates synthetic traces with excellent statistical properties. More than that SWIM can predict extremely well the performance of forwarding protocols, even the most sophisticated ones that base their mechanisms on the structure in communities of the network.

We have also shown how we can get larger networks with SWIM in a sound way. We have used this capability to perform the first experimental analysis of the scaling properties of several of the best forwarding protocols in the literature.

Acknowledgments

A. Mei was supported by a Marie Curie Outgoing International Fellowship funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 253461.

References

[1] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, "User Mobility for Opportunistic Ad-Hoc Networking," *Proc. IEEE Sixth Workshop Mobile Computing Systems and Applications (WMCSA '04)*, 2004.

[2] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05)*, 2005.

[3] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms," *Proc. IEEE INFOCOM*, 2006.

[4] T. Karagiannis, J.-Y.L. Boudec, and M. Vojnović, "Power Law and Exponential Decay of Inter Contact Times between Mobile Devices," *Proc. ACM MobiCom*, 2007.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: Real-World Mobility and Its Consequences for Opportunistic Forwarding," technical report, Computer Laboratory, Univ. of Cambridge, 2006.

[6] H. Cai and D.Y. Eun, "Crossing over the Bounded Domain: From Exponential to Power-Law Inter-Meeting Time in Manet," *Proc. ACM MobiCom*, 2007.

[7] P. Hui, E. Yoneki, S. Chan, and J. Crowcroft, "Distributed Community Detection in Delay Tolerant Networks," *Proc. IEEE/ACM Int'l Workshop Mobility Evolving Internet Architecture (MobiArch)*, 2007.

[8] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble Rap: Social-Based Forwarding in Delay Tolerant Networks," *Proc. ACM MobiHoc*, 2008.

[9] J. Whitbeck, V. Conan, and M. de Amorim, "Critical Analysis of Encounter Traces," *Proc. ACM Workshop Wireless of the Students, by the Students, for the Students (S3 '10)*, 2010.

[10] M. Musolesi and C. Mascolo, "Designing Mobility Models Based on Social Network Theory," *SIGMOBILE Mobile Computing Comm. Rev.*, vol. 11, no. 3, pp. 59-70, 2007.

[11] C. Boldrini, M. Conti, and A. Passarella, "The Sociable Traveller: Human Travelling Patterns in Social-Based Mobility," *Proc. ACM Seventh Int'l Symp. Mobility Management and Wireless Access (MobiWAC '09)*, 2009.

[12] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the Levy-Walk Nature of Human Mobility," *Proc. IEEE INFOCOM*, 2008.

[13] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "On Clustering Phenomenon in Mobile Partitioned Networks," *Proc. First ACM SIGMOBILE Int'l Workshop Mobility Models Networking Research*, 2008.

[14] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working Day Movement Model," *Proc. First ACM SIGMOBILE Int'l Workshop Mobility Models for Networking Research*, 2008.

[15] K. Lee, S. Hong, S.J. Kim, I. Rhee, and S. Chong, "SLAW: A Mobility Model for Human Walks," *Proc. IEEE INFOCOM*, 2009.

[16] C. Boldrini, M. Conti, and A. Passarella, "HCMM: Modelling Spatial and Temporal Properties of Human Mobility Driven by Users' Social Relationships," *Computer Comm.*, vol. 33, pp. 1056-1074, June 2010.

[17] A. Mei and J. Stefa, "SWIM: A Simple Model to Generate Small Mobile Worlds," *Proc. IEEE INFOCOM*, 2009.

[18] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking," *Proc. IEEE Seventh Ann. Comm. Soc. Conf. Sensor Mesh Ad Hoc Comm. Networks (SECON '10)*, 2010.

[19] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation Forwarding," *Proc. ACM MobiHoc*, 2008.

[20] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report CS-200006, Duke Univ., 2000.

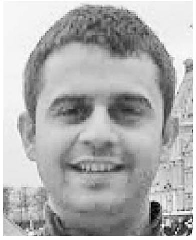
[21] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05)*, 2005.

[22] N.N. S.-Djukic, M. Pidrkowski, and M. Grossglauser, "Island Hopping: Efficient Mobility-Assisted Forwarding in Partitioned Networks," *Proc. IEEE Third Ann. Comm. Soc. Conf. Sensor Ad Hoc Comm. Networks (SECON '06)*, 2006.

[23] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi, "Understanding Individual Human Mobility Patterns," *Nature*, vol. 453, pp. 779-782, June 2008.

[24] C. Zhao and M. Sichitiu, "N-Body: Social Based Mobility Model for Wireless Ad Hoc Network Research," *Proc. IEEE Seventh Ann. Comm. Soc. Conf. Sensor Ad Hoc Comm. Networks (SECON)*, 2010.

[25] D. Fischer, K. Herrmann, and K.K. Rothermel, "GeSoMo: A General Social Mobility Model for Delay Tolerant Networks," *Proc. IEEE Seventh Int'l Conf. Mobile Ad Hoc Sensor Systems (MASS)*, 2010.
[26] A. Munjal, T. Camp, and W.C. Navidi, "Smooth: A Simple Way to Model Human Mobility," *Proc. ACM 14th Int'l Conf. Modeling, Analysis and Simulation Wireless and Mobile Systems (MSWiM '11)*, 2011.
[27] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "On the Levy-Walk Nature of Human Mobility," *IEEE/ACM Trans. Networking*, vol. 19, no. 3, pp. 630-643, June 2011.
[28] "SWIM: The Website," http://swim.di.uniroma1.it, 2013.
[29] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic Content Distribution in an Urban Setting," *Proc. ACM SIGCOMM Workshop Challenged Networks (CHANTS '06)*, 2006.
[30] J. Leguay, A. Lindgren, J. Scott, T. Riedman, J. Crowcroft, and P. Hui, "CRAWDAD Trace upmc/content/imote/cambridge (V. 2006-11-17)," http://crawdad.cs.dartmouth.edu/upmc/content/imote/cambridge, Nov. 2006.
[31] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD Trace cambridge/haggle/imote/infocom (V. 2006-01-31)," http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom, Jan. 2006.
[32] D. Kotz, T. Henderson, and I. Abyzov, "CRAWDAD Data Set dartmouth/campus (V. 2007-02-08)," http://crawdad.cs.dartmouth.edu/dartmouth/campus, 2013.
[33] H. Cai and D.Y. Eun, "Toward Stochastic Anatomy of Inter-Meeting Time Distribution under General Mobility Models," *Proc. ACM MobiHoc*, 2008.
[34] P. Hui, E. Yoneki, S.Y. Chan, and J. Crowcroft, "Distributed Community Detection in Delay Tolerant Networks," *Proc. IEEE/ACM Int'l Workshop Mobility Evolving Internet Architecture (MobiArch '07)*, 2007.
[35] E. Yoneki, P. Hui, S.Y. Chan, and J. Crowcroft, "A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks," *Proc. ACM Symp. Modeling, Analysis, and Simulation Wireless and Mobile Systems (MSWiM '07)*, 2007.
[36] D. Endres and J. Schindelin, "A New Metric for Probability Distributions," *IEEE Trans. Information Theory*, vol. 49, no. 7, pp. 1858-1860, July 2003.
[37] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," *Nature*, vol. 435, no. 7043, pp. 814-818, June 2005.



Sokol Kosta received the Laurea degree in computer science, summa cum laude, in 2009. He is currently working toward the PhD degree in the Computer Science Department at the Sapienza University of Rome, Italy. In 2010 and 2011, he visited T-Labs, Deutsche Telekom, Berlin, for a research internship. His research interests include mobile cloud computing, distributed systems, and analysis and modeling of social mobile wireless networks. He received the Best Paper Award for a PhD student from the Computer Science Department at Sapienza University in 2011.

Alessandro Mei received the Laurea degree in computer science, summa cum laude, from the University of Pisa, Italy, in 1994. He continued his studies as a PhD student in the Department of Mathematics at the University of Trento, Italy, and as a visiting scholar in the Department of Electrical Engineering-Systems at the University of Southern California during 1998 and part of 1999. He received the PhD degree in mathematics from the University of Trento in 1999. After a postdoctoral position at the University of Trento, he joined the faculty of the Computer Science Department at Sapienza University of Rome, Italy, in 2001. His primary research interests include computer system security and parallel, distributed, and networked systems. He is a past associate editor of the *IEEE Transactions on Computers* (2005-2009) and the general chair of IEEE IPDPS 2009, Rome, Italy. During 2010-2011, he was on leave at the Computer Science and Engineering Department, University of California, San Diego, supported by a Marie Curie fellowship. He is a member of the ACM and the IEEE.

Julinda Stefa received the Laurea degree in computer science, summa cum laude, and the PhD degree in computer science from the Sapienza University of Rome in July 2006 and February 2010, respectively. She is a postdoctoral researcher in the Computer Science Department at the Sapienza University of Rome, Italy. In 2005, she joined Google Zurich for three months as an engineering intern. She was a visiting scholar in the Computer Science Department at the University of North Carolina at Chapel Hill from November 2008 to April 2009 and a research intern at Microsoft Research, Cambridge, United Kingdom, from January to April in 2011. Her research interests include computer systems and network security, parallel and distributed systems, and analysis and modeling of social mobile wireless networks. She has published in some of the topmost conferences and journals like IEEE INFOCOM, ACM MobiHoc, IEEE ICDCS, *IEEE Transactions on Computers*, and *IEEE Transactions on Dependable and Secure Computing*. She received the Working Capital PNI research grant offered by Telecom Italia (30 winners out of 2,138). She is a member of the IEEE.