SAPIENZA
UNIVERSITÀ DI ROMA

OUTER SPACE, SELFISHNESS, AND INNOVATIVE SERVICES IN NETWORKS OF MOBILE INDIVIDUALS

by
Julinda Stefa

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the Degree of
DOCTOR OF PHYLOSOPHY in **COMPUTER SCIENCE**
at the
SAPIENZA UNIVERSITY OF ROME

December 2009

Thesis Committee

Prof. Alessandro Mei (First Member)
Department of Computer Science
Sapienza University of Rome, Italy

Prof. Roberto Di Pietro (Second Member)
Department of Computer Science
Sapienza University of Rome, Italy

Prof. Riccardo Silvestri (Third Member)
Department of Computer Science
Sapienza University of Rome, Italy

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Prof. Alessandro Mei Thesis Advisor

Approved for the University Committee on Graduate Studies.

External Reviewers

Prof. Levente Buttyán
Department of Telecommunications
Budapest University of Economics and Technology
Budapest, Hungary

Prof. Radha Poovendran
Department of Electrical Engineering
University of Washington
Seattle, USA

Dedikuar mamit dhe babit.

Acknowledgements

First and foremost I would like to thank my advisor, Prof. Alessandro Mei, for all his support during these years. He shared with me lots of excellent, elegant and inspiring ideas, and reinvented my way of fact-finding. He taught me everything ***from*** how to deal with difficulties, how to start thinking about new cool problems, how to approach them step by step, ***to*** how to achieve the highest goals. He always hold me to the highest standards and pushed me to do my best work all the time, yet reminding me every single moment, even in the toughest ones, of how fun research is! I'd really wish to become as skillful, enthusiastic, energetic, judicious, perspicacious, and outstandingly keen about both scientific research and teaching as Prof. Mei is, and, someday, to be able to command an audience and supervise students as awesomely as he can.

I want to express special gratitude to my High School Maths teacher Prof. Raimond Gumeni, for always believing in me.

I want to thank all my co-workers, phd colleagues, and all the people at the CS Department for all the wonderful moments passed during these three years.

A remarkable thank-you goes to the members of my internal thesis committee, Prof. Roberto Di Pietro and Prof. Riccardo Silvestri, and to my external reviewers, Prof. Levente Buttyán and Prof. Radha Poovendran, for the wonderful job they've done with my dissertation and for their excellent comments and advices on the matter.

Lastly, I want to thank my wonderful parents for supporting me during all my education studies, for always believing in me, and for teaching me that the best way of being ambitious is to be so with yourself. I'm blessed to have such awesome parents, and I'm really proud of them.

Contents

Introduction

As a kid, little did I know that I was living in the "era of internet", or that someday our live would have relied so much on a small fancy thing called cellphone. In the past 20 years we have seen the wireless system develop from first to second and third generation, providing more and more services that make our life easier. Now we can make phone calls, send emails, watch video clips, pay our bills, check our bank accounts, book plain tickets etc. All of this can be done while grabbing a coffee in our favorite bar, as long as there is a wireless service available provided by a network infrastructure. The further development of technology has made us even more eager. Now we dream about "Internet available every time and everywhere", even when we can not reach any wireless provider, maybe through *ad-hoc* connection to someone else's device that acts as a packet router to the access point in *his* proximity. This, and many other possible applications of ad-hoc wireless networks, where nodes act as routers for other nodes' packets in a decentralized way and without the need of a pre-fixed infrastructure, make this area very interesting and challenging in many ways.

Every single scenario of application of ad-hoc networks (e.g. natural disasters, military conflicts, surveillance etc.) presents new problems, often related to the diversity and limits of the devices involved therein. In any case though, researchers agree on the need for energy-efficient trustworthy protocols that allow for best management of resources in devices. Starting with routing, approaches that approximate shortest paths between source-destination pairs have become more than valuable in this context. An example is geographic routing, where the next relay is greedily chosen in order to reduce the distance from the destination's position.

Recent studies [7] have shown that if the area of deployment is a disk, uniform traffic

(source-destination pairs are uniformly distributed on the network) routed with shortest-path alike protocols induces congestion in the center of the network. Besides energy-efficiency problematics (nodes within congested areas tend to die earlier) congested areas bring also security issues. They weaken the network by giving more front to various types of attacks: Our experiments with geographic routing show that jamming a small area of 3% of the whole network blocks 25% of the whole traffic [1]. In our study we observe that this phenomena arises in every convex 2D surface, and that there exist *symmetric spaces* where the load of single nodes does not depend on their position within the network. Then we present *routing in outer space* that distributes equally the load among the network area. Although it consumes more energy globally with respect to geographic routing, it gets rid of the congested areas, makes the network more efficient in message delivery, and preserves longer its connectivity. In a word, routing in outer space yields a network that is more efficient, more secure, and that lives longer [2, 3].

We continue our study with another type of ad-hoc wireless network: The PSN (Pocket Switched Network) [8], where short range communicating devices are carried by humans. Network links are created and dropped in time, depending on the physical distance of device holders. In this scenario, end-to-end communication relies on contact opportunities among individuals and store-and-forward techniques. Because of the heterogeneity of human mobility, the design of forwarding protocols is a very challenging task. Although protocols based on flooding (e.g. Epidemic Routing [9]) are almost always able to find a path between source-destination pairs, they overload the network of message replicas by thus consuming more resources. Hence, protocols that aim to approximate Epidemic's performance and simultaneously limit forwarding costs like Delegation, Simbet, Bubble [10, 11, 12] have been presented in previous literature. All of these protocols rely on the cooperation among devices, without taking in consideration the inherent selfish nature of humans, the device holders. We investigate the issue, and show how badly Epidemic performs in presence of selfish individuals. The situation is even worse with more selective protocols such as Delegation. We attack the problem and come up with Give to Get Epidemic and Give to Get Delegation [5], two forwarding protocols that exploit *social aspects* of the network in detecting selfish behavior. Within the assumption that all the nodes are selfish, we show that both protocols are Nash Equilibria. Our extended experiments with

real mobility traces show that the detection rate of both protocols is higher than 90% [5]. They induce an extremely small delay overhead while reducing of more than 20% the number of message replicas with respect to their vanilla alter egos. We test our protocols also in presence of a natural variation of selfishness–nodes that are selfish with outsiders and faithful with individuals from the same community. Even in this case our protocols are shown to be very efficient in detecting misbehaviors.

Validating protocols for mobile social networks is not a simple task. If real mobility traces are lacking, or are not adequate for the scenario considered (e.g. studies on disease spreading in countries need large-scale data), a mobility model has to be used. Since when the RWP (Random Way Point) model [13] was shown to be inadequate in modeling human mobility [14, 15, 16], the design of a suitable mobility model for humans have attracted many researchers. Previous attempts are either complicated and hard to implement, or are not able to predict accurately performance of protocols. In a first endeavor to solve the problem we present SWIM (Small World In Motion) [4], a model based on simple observations of everyday life: 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, which we believe to be the base of human mobility in real life. SWIM is the first model shown experimentally and theoretically to have the statistical properties of human mobility. Moreover, SWIM can predict well the performance of forwarding protocols: Epidemic and Delegation Forwarding perform the same on SWIM and on the real traces, in terms of success delivery, delay and cost of message replicas. These results make SWIM a useful tool in the study of better understanding human mobility with the aim to design suitable and innovative services and protocols for social mobile networks.

Inspired by SWIM, we push our study of human mobility even further, putting the network individual and his needs in the center of our work. For this purpose we introduce the concept of INdividual MObile NETwork (IMONET) [6], where each individual is associated with a *community profile* that characterizes him with a certain set of habits, interests and social relations within the network. We then use this characterization in the design of two novel communication paradigms *interest* and *community* casting. In interest-casting,

a message characterized by a certain *relevance profile* is selectively forwarded to potentially interested IMONETs members, while in community-casting a message is selectively forwarded to the IMONET nodes who are members of a certain community. We then present the complete design of an interest- and community-casting protocol called CIF [6], and show through extensive simulations that CIF is able to disseminate information to interested IMONET members almost as quickly as Epidemic Forwarding, but with a much smaller communication overhead.

In the attempt to better explain our findings in the huge area of wireless ad-hoc networks we start from the *outer space*.

Chapter 1

Routing In Outer Space

During the past years the interest in multi-hop wireless networks has grown significantly. Their ability to use nodes as a relay to deliver messages from source to destination makes these networks not only scalable but also usable in various areas and contexts. One of the most representative and important examples of multi-hop wireless networks are wireless sensor networks where small devices equipped with a radio transmitter and a battery are deployed in an geographic area for monitoring or measuring of some desired property like temperature, pressure, or others [17, 18]. The limited resources and capacities of the nodes in such networks turn routing into a very interesting and challenging issue. Protocols that approximate shortest paths between source-destination pairs are preferred to others for being more efficient in terms of overall energy consumed per message.

In this chapter we consider security-related and energy-efficiency issues in multi-hop wireless networks. We start our work from the observation, known in the literature, that shortest path routing creates congested areas in multi-hop wireless networks. These areas are critical—they generate both security and energy efficiency issues: Our experiments show that hot-spots of just 3% of the network area handle 25% of the messages. A smart attacker would not lose the chance to jam in such conditions: Low cost (targeted to a small surface), high damage. Moreover, the nodes within the hot-spots begin to die early, exacerbating the phenomena to other nodes outside this area. The network looses quickly its routing capacity and becomes useless. Lastly, if the initial position of the nodes is to

be chosen individually (students in a campus with intermittent wireless connection, auto-positioning sensors in an area of interest etc.) well known to-be hot-spot regions would remain empty. This unwanted arbitrary distribution of nodes might result in a disconnected network.

We attack these problems and set out routing in outer space, a new routing mechanism that transforms any shortest path routing protocol (or approximated versions of it) into a new protocol that does not create congested areas, does not have the associated security-related issues, and does not encourage selfish positioning. Moreover, the network is more energy-efficient than the same network using the original routing protocol (in spite of using more energy globally), and dies more gracefully. We also describe applications of our idea to mobility and to a security protocol for the detection of node replication attacks.

The results presented in this chapter appear in [1, 2, 3].

1.1 Routing and congestion in multi-hop wireless networks

Routing in multi-hop wireless networks is one of most important, interesting, and challenging problems due to network device limitations and network dynamics. As a matter of fact this is one of the most studied topics in this area, and the literature on routing protocols for multi-hop wireless networks is vast. There have been proposed various proactive protocols [19, 20, 21] that maintain routes continuously, reactive protocols [22, 23, 24] that create routes on demand or a hybrid [25]. For a good survey and comparison see [26, 27].

Proactive protocols are usually based on the distance vector routing technique, thus on storing, updating and broadcasting routing tables from the nodes. As an example, the DSDV (Destination-Sequenced Distance-Vector Routing) protocol [19] is a hop-by-hop distance vector routing protocol where nodes have to broadcast their routing tables periodically. Each entry in the routing table is given a sequence number that enables nodes to distinguish between stale routes and new ones. Unlike the vanilla distance vector technique, it comes with the advantage of being loop-free.

Reactive protocols create routes on-demand, thus routing information to a certain destination becomes available only when needed. The most representative protocol of this group is the DSR (Dynamic Source Routing) [22] protocol. It is based on the concept of

source routing where nodes maintain cache route information. When a node needs to send a message it first checks whether an unexpired route for that destination is present in the cache. If this is not the case, the node triggers a route discovery protocol for the destination. A route reply is generated when either the destination or an intermediate node containing in its cache an unexpired route to the destination is reached.

Both these routing techniques have their drawbacks: Proactive protocols rely in underlying mechanisms that create and maintain routing tables involving constant and periodic broadcasting of routing information. Not only nodes dedicate a non negligible amount of memory to the permanent storage of this information, but also they have to deal with the signal traffic incurred by this mechanism. On the other hand, in on-demand routing protocols nodes routing informations is not always available. Nodes often have to wait till route establishment. Moreover these protocols also incur signal traffic due to the propagation of route requests and route replies each time a new destination needs to be reached or an old route has expired. Since nodes have limited resources in terms of memory, processing capacity and battery, it is unacceptable that most of these resources is spent in routing. Thus energy-efficient protocols that need less information possible become more than valuable in this context.

Geographic or position-based routing [28] where nodes use location information of their neighbors in choosing the next relay to destination seem to be one of the most feasible and studied approach. Nodes need to maintain a minimum state, and the overhead incurred is very low. In geographic routing nodes need to know their position within the network, and the position of their neighbors. This can be obtained through any of the localization systems present in the literature: GPS (Global Positioning System), infrastructure-based localization systems [29, 30], ad-hoc localization systems [31, 32] or secure localization techniques [33, 34]. When a source node wants to send a packet to a destination it chooses as next relay the neighbor that allows for the greatest progress towards the destination's geographic position. Examples of research work on this approach are protocols like GEAR [35], GAF [36], localization error-resilient version of geographic routing [37], and geographic routing for mobile systems [38, 39]. For a good starting survey see [40].

All these protocols try to approximate the shortest path between source and destination over the network. In [41], the authors analytically study the impact of shortest single-path

routing on node traffic load by approximating single paths to line segments, and show that multi-path routing, although introducing a larger overhead, provides better congestion and traffic balancing. Further work in the same direction [42] shows that multi-path routing can balance load only if a very high number of paths is used.

In [7], the authors analyze the load for a homogeneous multi-hop wireless network for the case of straight line routing. They show that relay traffic induces congested areas. If the traffic pattern is uniform, i. e. every message has a random source and a random destination uniformly and independently chosen, and the network area is a disk, then the nodes at the center of the disk have to relay much more messages that the other nodes. Of course, geographic routing (which, in dense networks, approximates the shortest path between source and destination) also suffers from the same problems. Moreover, the same phenomena arises if the network area is a square, a rectangle, or any other two-dimensional convex surface. As we will see later on this chapter, our experiments show that, when using geographic routing on a network deployed in a square, 25% of the messages are relayed by the nodes in a small central congested region whose area is 3% of the total area of the square.

Congested areas are bad for a number of important reasons. They raise security-related issues: If a large number of messages are relayed by the nodes deployed in a relatively small congested region, then jamming can be a vicious attack. It is usually expensive to jam a large geographical area, it is much cheaper and effective to jam a small congested region. In the square, for example, it is enough to jam 3% of the network area to stop 25% of the messages. Moreover, if an attacker has the goal of getting control over as many communications as possible, then it is enough to control 3% of the network nodes to handle 25% of the messages.

There are also energy-efficiency issues: Aside from re-transmissions, that are costly and, in congested areas, more frequent, the nodes have to relay a much larger number of messages. Therefore the nodes in these areas will die earlier than the other nodes in the network, exacerbating the problem for the nodes in the same area that are still operational. In the long run, this results in holes in the network and in a faster, and less graceful, death of the system.

Lastly, there may be other concerns in the contexts where the nodes are carried by individual independent entities. In this chapter we do not consider mobility. However, if the position of the node can be chosen by the node in such a way to maximize its own advantage, and if energy is an issue, then every node would stay close to the border, where it can get the same services without having to relay other nodes' messages. If the nodes are selfish, an uneven distribution of the load in the network area leads to an irregular distribution of the nodes—there is no point in positioning in the place where the battery is going to last the shortest. Selfish behavior is a recent concern in the networking community and it is rapidly gaining importance. Most mechanisms proposed in the literature [43, 44, 45] can be used to force selfish nodes to execute the protocol truthfully, wherever they are positioned, but they do not help in preventing selfish positioning or moving. For the best of our knowledge, here we are raising a new concern, that can be important in mobile networks or whenever the position of the node can be an independent and selfish choice, like in networks of individuals (e. g. students in a university campus network).

The problem of reducing congestion at the center of a network deployed in a disk in the case of uniform traffic has been considered in [46]. The authors consider a number of possible heuristics like selecting routes along inner and outer radii and switching between them at a random point, moving between the radii linearly, and so on. Later, and independently of this work, the same issues are addressed in [47]. The authors present a theoretical approach to solve the problem showing that an optimum routing scheme based on shortest paths can be expressed in terms of geometric optics and computed by linear programming. Being the optimal trajectories they find not expressible by closed form formulas, hence not applicable in practice, they also present a practical solution that approximates the optimum. This solution is shown to be implementable and close to the optimum in the case of the disk, while its performance is not as good in the case of the square.

A lot of work has been done regarding energy-efficiency issues, and several approaches try to solve the problem locally, like [35, 48, 49]. These approaches are useful to balance the load reactively and to smooth the energy level among neighbors, while they cannot remove congested areas.

Dealing with congestion and other issues that it brings—security, energy-efficiency, and tolerance to (a particular case of) selfish behavior—is an important and non-trivial problem. We attack this problem and set out *routing in outer space*, a new routing mechanism that transforms any shortest path routing protocol (or approximated versions of it) into a new protocol that, in case of uniform traffic, guarantees that the network does not have congested areas, does not have the associated security issues, and, in spite of using more energy globally, lives longer than the same network using the original routing protocol—that is, it is more energy-efficient. We support our claims by showing routing in outer space based on geographic routing, and by performing a large set of experiments.

1.2 The square is not symmetric

In this section we try to explore the reasons behind hot-spots creation in case of large multi-hop wireless networks where shortest-path alike protocols are used to route uniform traffic pattern. We come up with a definition of *symmetric spaces* where this problem does not arise, show how all bi-dimensional surfaces are not symmetric in the sense of our definition, and propose a transformation to asymmetric spaces in symmetric ones.

We model the multi-hop wireless network as a undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The nodes are ad-hoc deployed on the network area S. Formally, it is enough to assume that S is a metric space with distance d_S and that every node is a point on S. Given two nodes $u, v \in V$ deployed on S, we will denote the distance between their positions on the space with $d_S(u, v)$. The nodes have a transmission range r—two nodes $u, v \in V$ are connected by a wireless link $uv \in E$ if $d_S(u, v) \leq r$, that is, their distance is at most r. The common practice in the literature is to take a convex surface as S, usually a square, a rectangle, or a disk, with the usual Euclidean distance. In our study we assume that the nodes know their position, either by being equipped with a GPS unit or by using one of the many localization protocols [31, 32], and that they know the boundaries of the network area S; this is possible either by pre-loading this information on the nodes before deployment, or by using one of the protocols in [50, 51, 52].

We started from the observation that shortest path routing on the square, or even an approximate version of it, generates congested areas on the center of the network. We have

already discussed that this phenomenon is not desirable. The same problem is present on the rectangle, on the disk, and on any two dimensional convex deployment of the network, which is the common case in practice. Our idea is to get rid of congested areas by relinquishing shortest paths. As the first step, we have to realize that there do exist metric spaces that do not present the problem. First, we need a formal definition of the key property of the metric space we are looking for.

Definition 1.2.1. *Consider a multi-hop wireless network deployed on a space S. Fix a node u and choose its position on S arbitrarily. Then, deploy the other nodes of the network uniformly and independently at random. We will say that S is* symmetric *if, chosen two nodes v_1 and v_2 uniformly at random in the network, the probability that u is on the shortest path from v_1 to v_2 does not depend on its position.*

The disk is not a symmetric space as in the above definition. It has been clearly shown in [7]—if node u is on the center of the circle or nearby, the probability that u is traversed by a message routed along the shortest path from a random source node v_1 to a random destination v_2 is larger than that of a node away from the center of the network area. Clearly, the square has exactly the same problem. This claim is confirmed by our experiments: 25% of the shortest paths traverse a relatively small central disk whose area is 3% of the entire square.

To solve these problems, our idea is to map the network nodes onto a symmetric space (the *outer space*) through a mapping that preserves the initial network properties (such as distribution, number of nodes, and, with some limitations, distances between them). Note that there is no need that the mapping be continuous (actually, restricting to continuous mappings would make our idea lose most of its interest). The second step is to route messages through the shortest paths as they are defined on the outer space. When the outer space and the corresponding mapping are clear from the context, we will call these paths the *outer space shortest paths*. Since the outer space is symmetric, we can actually prove that every node in the network has the same probability of being traversed by an outer space shortest path, on average. In the following section we will see that, based on this idea, we can design practical routing protocols that do not have highly congested areas. Furthermore, the routing protocol that we will present prolongs considerably the network

lifetime. Now, let's take a step back and proceed formally.

1.2.1 Routing in outer space

Let S be the original space where the network is deployed, and let T be the outer space, an abstract space we use to describe routes, both metric spaces with respective distances d_S and d_T. We are looking for a mapping function $\phi : S \mapsto T$ with the following properties:

1. if u is a point taken uniformly at random on S, then $\phi(u)$ is also taken uniformly at random on T;

2. for every $r > 0$, $u, v \in S$, $u \neq v$, if $d_T(\phi(u), \phi(v)) \leq r$ then $d_S(u, v) \leq r$.

Property 1 guarantees that a uniform traffic on S is still a uniform traffic when mapped onto T through ϕ, and Property 2 says that paths on T are paths on S, when the nodes are mapped into T using ϕ. Later we will see why these properties are important.

Definition 1.2.2. *A mapping $\phi : S \mapsto T$ is* fair *if it enjoys Properties 1 and 2.*

Once such a fair mapping has been fixed, any message from node u to node v can be routed following a shortest path between the images of u and v and through the images of some of the network nodes under ϕ on space T. Let $\phi(u), \phi(w_1), \phi(w_2), \ldots, \phi(w_h), \phi(v)$ be such a path. Being ϕ a fair mapping, the path $u, w_1, w_2, \ldots, w_h, v$ is a well defined path on S. Indeed, any two consecutive nodes in the shortest path on T are neighbors in S as well, thanks to Property 2. If T is *symmetric* as in Definition 1.2.1, the routing through ϕ will be well distributed over T, since ϕ has Property 1. Hence, this path can be used to route messages on S, giving as a result a homogeneous distribution of the message flow over all the original network area.

Theorem 1.2.1. *Let $\phi : S \mapsto T$ be a mapping from source metric space S to target metric space T. Assume that ϕ is fair and T is symmetric. Fixed a node $u \in S$, deployed the other nodes of the network uniformly at random, and taken a source $v_1 \in S$ and a destination $v_2 \in S$ uniformly at random, the probability that the outer space shortest path from v_1 to v_2 defined by ϕ traverses u is independent of the position of u on S.*

The above theorem shows how to build a routing protocol on a not symmetric network area, in such a way that the message flow is distributed homogeneously over all the network. What is needed is to determine a symmetric space (the outer space) and a fair mapping for it, and then to "transform" the shortest paths on the original network area into the corresponding outer space shortest paths.

We assume that the original network area is a square of side 1. An excellent candidate as a symmetric outer space is the torus. A *torus* is a 2D manifold in 3D that we can model as $T = [0,t] \times [0,t]$. Let u_x and u_y be the coordinates of the position of node u on the torus. We can endow T with the following distance d_T:

$$d_T(u,v) = \sqrt{d_x^2 + d_y^2}, \tag{1.2.1}$$

where

$$d_x = \min\{|u_x - v_x|, t - |u_x - v_x|\}, \text{ and} \tag{1.2.2}$$

$$d_y = \min\{|u_y - v_y|, t - |u_y - v_y|\}. \tag{1.2.3}$$

The common way to visualize a torus is to consider a square, and then to fold it in such a way that the left side is glued together with the right side, and that the top side is glued together with the bottom side. In the following, we will picture the torus unfolded, just like a square, as it is commonly done to easily see this space in 2D.

Fact 1.2.1. *A torus is symmetric as in Definition 1.2.1.*

Clearly, virtually no wireless network in real life is deployed on a torus. Here, we are using the torus just as an abstract space. We are *not* making any unreasonable assumption on the nodes of the network being physically placed on a torus like area with continuous boundaries, nor are we assuming that the network area becomes suddenly a torus. Indeed, we assume that the real network is deployed on the square, where the nodes close to one side *cannot* communicate with the nodes close to the opposite side. Crucially, the paths used to deliver the messages are computed as they are defined through a fair mapping onto the torus, the outer space. Coming back to our idea, now that the target symmetric outer space has been chosen, what is left to do is to find a fair mapping ϕ_{ST} from the square to

the torus.

Let $S = [0,1] \times [0,1]$ be a square, and let $T = [0,2] \times [0,2]$ be a torus. We propose to define ϕ_{ST} as follows: $\phi_{ST}((x,y)) = (x',y')$ where:

$$x' = \begin{cases} x & \text{with probability } 1/2, \\ 2-x & \text{with probability } 1/2, \end{cases}$$

and

$$y' = \begin{cases} y & \text{with probability } 1/2, \\ 2-y & \text{with probability } 1/2. \end{cases}$$

Even though ϕ_{ST} is partly probabilistic, this does not mean that routing in outer space is a *random* routing scheme, like sending the packet along a Brownian path or like sending the packet to a random intermediate (an idea that has been used a lot in routing in parallel architectures and, later, also in network routing). Indeed, it is pretty easy to come up with a very similar completely deterministic version of ϕ_{ST} with exactly the same properties, for our purposes. This deterministic version, however, is more complex to describe and to deal with, and this is the sole motivation to choose a partly probabilistic, and technically simpler, version.

An example of ϕ_{ST} can be seen in Figure 1.1, where a node on the square is mapped to one of the four equally probable images on the torus.

Theorem 1.2.2. *ϕ_{ST} is a* fair *mapping.*

Proof. We show that ϕ_{ST} has both Property 1 and 2.

Let $F \subset S$ be defined as follows:

$$F = \{(x,y) | (x = 0 \vee x = 1 \vee y = 0 \vee y = 1\}.$$

Set F is the set of the points on the borders of the four quadrants shown in the torus of Figure 1.1. Since F is a measure-zero set, we can prove Property 1 in $T \setminus F$ without loss of generality.

Consider a point $z \in T \setminus F$. Point z lies in one of the four quadrants of the torus shown in Figure 1.1—it can not be on the border. Therefore, it is possible to find $\varepsilon > 0$ small

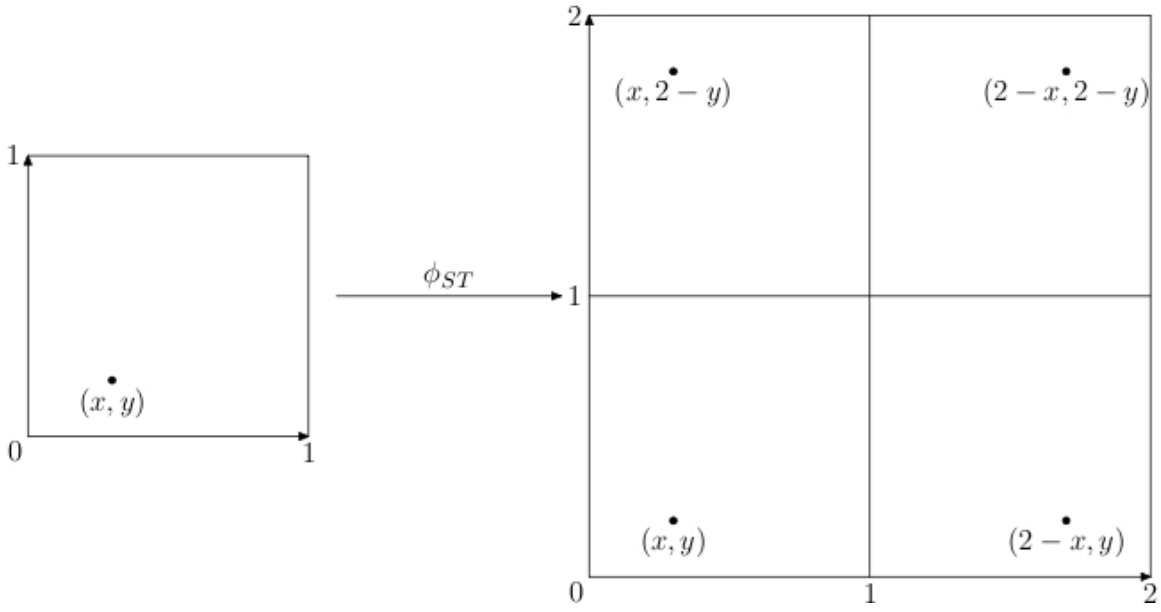


Figure 1.1: Example of mapping a point from the square to the torus through ϕ_{ST}. Point (x,y) on the square $S = [0,1] \times [0,1]$ has four possible and equally probable images on the torus $T = [0,2] \times [0,2]$. According to ϕ_{ST}, only one of the images will actually appear on T.

enough, for which the ball $B(z,\varepsilon)$ with center z and radius ε, is completely contained in the same quadrant. To prove Property 1, we need to show that, if we take u uniformly at random in S, then the probability that $\phi_{ST}(u) \in B(z,\varepsilon)$ is $\pi\varepsilon^2/4$, just proportional to the area of $B(z,\varepsilon)$ in $T \setminus F$ (recall that the area of $T \setminus F$ is 4). This shows that $\phi_{ST}(u)$ is taken uniformly at random as well.

Assume that point z lies in the quadrant on the upper-right corner. The probability that $\phi_{ST}(u) \in B(z,\varepsilon)$ is equal to the probability that u is chosen in $B(z',\varepsilon) \subset S$, where $z'_x = 2 - z_x$ and $z'_y = 2 - z_y$, and that ϕ_{ST} maps u to the quadrant on the upper-right corner. The first event happens with probability $\pi\varepsilon^2$, since u is taken uniformly at random in S, $B(z',\varepsilon) \subset S$ has area $\pi\varepsilon^2$, and S has area 1. The second event happens with probability $1/4$. The two events are independent, and therefore the probability that $\phi_{ST}(x) \in B(z,\varepsilon)$ is equal to $\pi\varepsilon^2/4$, as claimed. The case when z lies in one of the other three quadrants can be dealt similarly.

To show Property 2, we can prove it separately for each axis. We can prove the following: Let $r \in \mathbb{R}$, $r > 0$, and let $u, v \in S$, $u \neq v$. If $d_T^x(\phi(u), \phi(v)) \leq r$ then $d_S^x(u,v) \leq r$, where d^x denotes the distance along the x-axis. If ϕ_{ST} makes the same random choice for both u and v (that is, either both x-coordinates are not changed, or both are reflected), than the claim is just trivial. Assume that ϕ_{ST} does not make the same random choice for both u and v. For example, say that ϕ_{ST} does not change the x-coordinate of u and reflects the x-coordinate of v. That is, $\phi_{ST}^x(u) = u_x$ and $\phi_{ST}^x(v) = 2 - v_x$, where ϕ_{ST}^x is the projection of ϕ_{ST} along the x-axis. If $d_T^x(\phi(u), \phi(v)) \leq r$, then either $|(2 - v_x) - u_x| \leq r$ or $2 - |(2 - v_x) - u_x| \leq r$. In both cases, it is easy to show by elementary algebra that $d_S^x(u,v) \leq r$, as claimed. Exactly in the same way we can see that the claim holds for the y-axis. Therefore, Property 2 holds as well. □

It is interesting to note that even if two points are neighbors on the square, they might not be neighbors on the torus when mapped through ϕ_{ST}. Generally speaking, it is impossible to build a mapping with both this property and Property 2, since the square and the torus are topologically different.

The outer space shortest path between two nodes may be different from the corresponding shortest path. Clearly, it can not be shorter by definition of shortest path on S. A natural question to ask is whether we can bound the stretch, that is, how much longer may the outer space shortest path be compared with the corresponding shortest path? Unfortunately, the answer is that the stretch cannot be bounded by a constant. However, quite surprisingly, we can prove a very good constant bound in the case when many messages are sent through the network, that is the common case in practice. Indeed, while in the worst case the stretch can be high, it is not on average if we assume a uniform traffic. This claim is formalized in the following theorem, where we show that, on expectation, the distance of the images under ϕ_{ST} of two nodes taken uniformly and independently at random is at most the double of the original distance.

Theorem 1.2.3. *If nodes u, v are taken uniformly at random on the square $S = [0,1] \times [0,1]$, and $\phi_{ST}(u), \phi_{ST}(v)$ are their respective images under ϕ_{ST} on the torus $T = [0,2] \times [0,2]$, then*

$$E[d_T(\phi_{ST}(u), \phi_{ST}(v))] \leq 2E[d_S(u,v)].$$

Proof. Let $u, v \in S$ be two nodes whose position is taken uniformly at random, and let $E[d_S(u,v)] = \mu$ be the expectation of their distance on S. Since ϕ_{ST} is fair, also $\phi_{ST}(u)$ and $\phi_{ST}(v)$ are taken uniformly at random in the torus. Clearly, the distance between $\phi_{ST}(u)$ and $\phi_{ST}(v)$ on the torus cannot be larger of the distance of $\phi_{ST}(u)$ and $\phi_{ST}(v)$ on a square $S' = [0,2] \times [0,2]$. Indeed, every path on the torus is also a path on the square (the opposite is not true); and the average distance of two random points in a square of edge two is the double of the average distance of two random points in a square of edge one. Therefore,

$$\begin{aligned} E[d_T(\phi_{ST}(u), \phi_{ST}(v))] &\leq E[d_{S'}(\phi_{ST}(u), \phi_{ST}(v))] \\ &= 2E[d_S(u,v)] \\ &= 2\mu. \end{aligned}$$

□

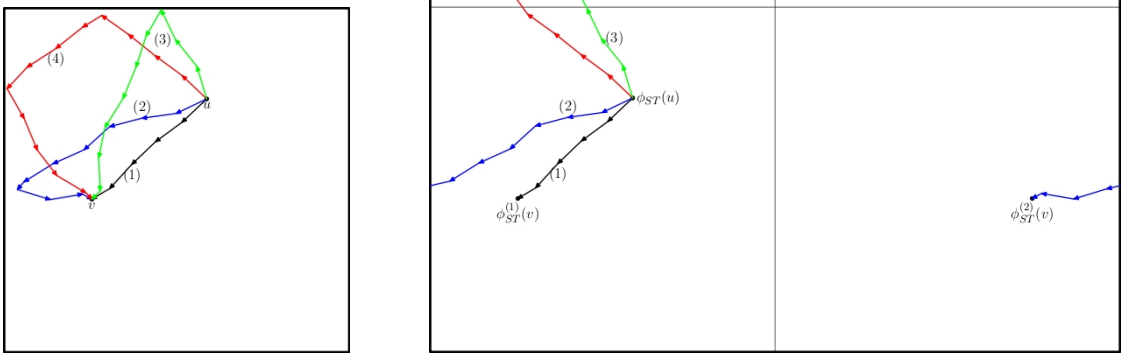
In the following, we will see with experiments that the actual average stretch is even smaller.

Of course, it is always possible to use the outer space shortest path only when the stretch of that particular path is small, and to use the classical shortest path when the stretch is high and the outer space shortest path is going to cost a lot more. However, we do not perform this kind of optimizations—even though they may reduce the global energy required by the network to deliver the messages, they also unbalance the load among the nodes. Therefore, we want to consider routing in outer space in its cleanest version. In the following, we will implement our idea in a practical routing protocol derived from geographical routing, and show its performance by means of experiments.

1.3 Routing in outer space in practice

We start from geographic routing, a simple protocol that, when the network is dense enough, can be shown to approximate shortest path routing quite well [43]. Here, we define *outer space geographic routing*, its outer space counterpart.

In geographic routing, the destination of a message is a geographical position in the



(a) Four equally probable outer space geographic routings between node u and node v.

(b) Four equally probable outer space geographic routings between one of the possible image of node u, $\phi_{ST}(u)$, and the four possible images of v, $\phi_{ST}^{(1)}(v), \ldots, \phi_{ST}^{(4)}(v)$.

Figure 1.2: Assume, without loss of generality, that $\phi_{ST}(u)$ is fixed. Subfigure (b) shows the four equiprobable shortest paths from $\phi_{ST}(u)$ to the four possible $\phi_{ST}(v)$. Subfigure (a) shows the corresponding four equiprobable outer space shortest paths. Path (1) is just the traditional geographic routing between u and v. The network used to build this example is made of $6,441$ nodes. (If you choose another image for $\phi_{ST}(u)$, the shortest paths are moved in the torus without changing the corresponding outer space shortest paths.)

network area—in the square in our case. Every relay node performs a very simple protocol: Send the message to the node that is closer to destination. If such a node does not exist, then the message is delivered. If the network is dense, every message is delivered to the node closest to destination. It is known that this simple version of geographic routing sometimes is not able to deliver the message to the node closest to destination, and there are plenty of ways to overcome this problem in the literature. However, we do not consider these extensions (outer space geographic routing could as well be based on these more complex and complete versions), since the increased complexity do not add much to this work.

Outer space geographic routing works quite as simply. Every relay node looks at the destination x of the message, and forwards it to the node u that minimizes the distance $d_T(\phi_{ST}(x), \phi_{ST}(u))$. Just like geographic routing, implemented on the outer space.

Take, as an example, a message from node u destined to a geographic position close to node v. According to the definition of ϕ_{ST}, each node on the square S has four possible and equally probable images on the torus T. This implies that for each pair u, v of nodes on S there are four possible and equally probable pairs of images $\phi_{ST}(u)$, $\phi_{ST}(v)$ on T. (Actually, there are 16 possible and equiprobable such couples, which fall into 4 different classes of symmetry up to isomorphism.) This yields four possible and different outer space geographic routes between the images u and v under ϕ_{ST}. Hence, between any two nodes on the square there is one out of four different and equally probable outer space routes. To see an example of the four routes, see Figure 1.2.

To implement such a routing, it is enough that the nodes know their position in the square. Then, computing ϕ_{ST} for itself and the neighbors is trivial and fast. Note that it is not really important that the nodes agree on which of the four possible images is actually chosen for any particular node (except for the destination, but the problem can easily be fixed). However, to get this agreement for every node it is enough to compute ϕ_{ST} by using the same pseudo-random number generator, seeded with the id of the node being mapped.

Note that the mapping makes the graph of the network sparser as neighbors in the original network may not be neighbors in the outer space. Thus greedy forwarding may have a higher chance to get stuck at a dead end. Whenever this is a problem, we can implement the protocol by assuming that all of the four images of every node are present in the outer space, simultaneously. In this way, the network does not lose density while all the benefits of routing in outer space are preserved. This is what we have done in all of the experiments.

1.3.1 Node and network properties, assumptions and simulation environment

We model our network node as a sensor. An example can be the Mica2DOT node (outdoor range 150m, 3V coin cell battery). These nodes have been widely used in sensor network

academic research and in real testbeds. We use these devices for our experiments as a well-known energy model. However, we expect that the results are meaningful for ad-hoc networks based on other devices as well, after proper scaling.

For our experiments, we have considered networks with up to 10,000 nodes, distributed using a Poisson process on a square of side $1,500$m. In the following, we will assume for the sake of simplicity that the side of the square is 1, and that the node transmission range is 0.1.

We inject a *uniform traffic* in the network—every message has a random source and a random destination uniformly and independently chosen. This type of traffic distribution is highly used in network simulations, for example when the goal is to study network capacity limits, optimal routing, and security properties [53, 54, 55]. We assume that the nodes know their position on the network area. Therefore, they need to know both their absolute position, and their position within the square. The nodes can get the absolute position either in hardware, by using a GPS (Global Positioning System), or in software. There exist several techniques for location sensing like those based on proximity or triangulation using different types of signals like radio, infrared acoustic, etc. Based on these techniques, several location systems have been proposed in the literature like infrastructure-based localization systems [56, 57] and ad-hoc localization systems [31, 32]. In [58] you can find a survey on these systems while in [59] the authors present NoGeo: A location system that permits routing based on virtual positions of nodes.

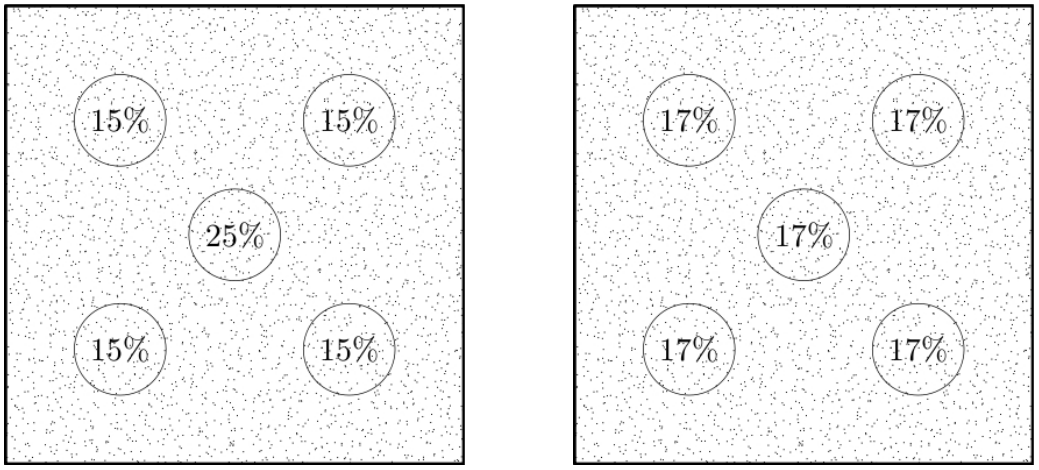
Once the absolute position is known, we can get the nodes to know their relative position within the square by pre-loading the information on the deployment area, or by using one of the several techniques for boundary detection based on geometry methods, statistical methods, and topological methods (see [50, 51, 52]).

In the next two sections we present the results of the experiments we have performed, comparing our routing scheme with geographic routing over the same networks and with the same set of messages to route. For the experiments we have used our own event-based simulator. The assumptions and the network properties listed above have been exactly reflected in the behavior of the simulator.

1.3.2 Security-related experiments

In these experiments, we measure the number of messages whose routing path traverses five sub-areas of the same size in the network area. Every sub-area is a circle of radius 0.1 (incidentally, the same of the transmission radius of a network node), that corresponds to an area of 3.14% of the whole network surface. The sub-areas are centered in some "crucial" points of the network area: The center and the middle-half-diagonals points. The center of the network is known to be the most congested area. We want to test whether the middle-half-diagonal centered areas handle a significantly smaller number of messages. More specifically we consider the sub-areas centered in the points of coordinates $(0.5, 0.5)$, $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$, $(0.75, 0.75)$, assuming a square of side one. Our experiments are done on networks with different number of nodes (from 1,000 to 10,000). For each network we have run both geographic routing and outer space geographic routing on message sets of different cardinality (from $50,000$ to $1,000,000$ messages, generated as an instance of uniform traffic). In Figure 1.3 we present the average of the results obtained with a network of $1,336$ nodes generated by a Poisson process, but we stress that exactly the same results are obtained for networks with up to $10,000$ nodes. As it can be seen, the experiments fully support the findings in [7]. Geographic routing (see Figure 1.3(a)) concentrates a relevant fraction of the messages on a small central area of the network, while the other sub-areas handle on average little more than the half. We have already discussed why this is dangerous, and important to avoid. Figure 1.3(b) shows the result with the same set of messages and the same network deployment, this time using outer space geographic routing. The message load in the central sub-area is 32% lower compared with the load of the same sub-area in the case of the geographic routing. Outer space geographic routing seems to transform the network area in a symmetric surface, making sure that the number of message handled by all the sub-areas remains reasonably low, 17%, and equally distributed. As a result, the load among the network nodes is equally balanced and there are no "over-loaded" areas. This network is intuitively stronger—there are no areas that are clearly more rewarding as objective of a malicious attack, and no areas that have more "responsibilities" than others.

Furthermore, Figure 1.3(a) clearly shows that, with geographic routing, it is not a good



(a) Geographic routing. (b) Outer space geographic routing.

Figure 1.3: The average fraction of the messages whose routing path traverses the selected sub-areas of a network of 1,336 nodes, in the case of geographic routing and in the case of outer space geographic routing.

strategy to stay in the center of the network if you want to save your battery. If the nodes are selfish, it is a much better strategy to position in one of the sub-central areas, for example, where the battery is going to last 66% longer. Even better if you move towards one of the corners of the square, where there is virtually no traffic to relay. Conversely, when using outer space geographic routing, there is no advantage in choosing any particular position, since the relay traffic is equally distributed everywhere.

1.3.3 How to live longer by consuming more energy

In this section we present the experiments related to energy-efficiency. What Theorem 1.2.3 says in a sentence is that the paths used by outer space geographic routing are on average (at most) twice as long as the paths used by geographic routing. This should have an immediate consequence on energy consumption: Messages routed with outer space geographic routing should make the network nodes consume more energy, up to twice as much. And actually it is so. What it turns out with our experiments is that using routing in outer space the average path stretch is 1.34. Even though this translates into a 34% larger global energy consumption, we will see that, in addition to better security and absence of congested areas,

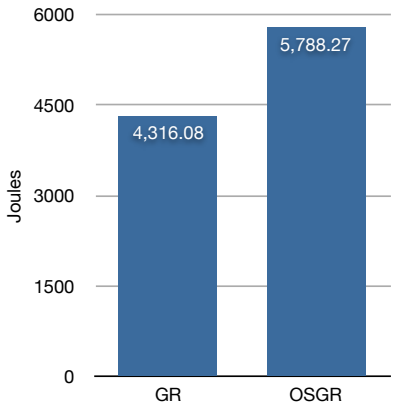
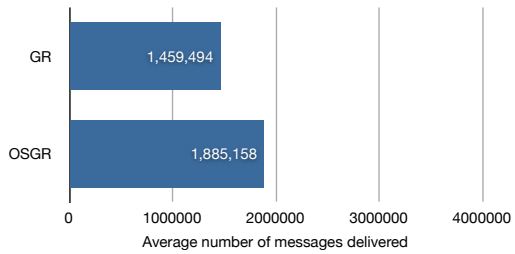


Figure 1.4: Global energy consumption of the network after running geographic routing (GR) and outer space geographic routing (OSGR) on sets of 50,000 messages each. The network is made of 1,625 nodes.

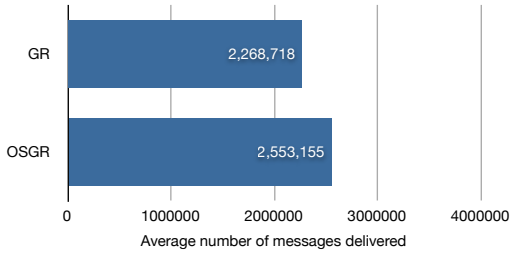
the network has also excellent benefits from an energy-efficiency point of view when using routing in outer space. Figure 1.4 shows the global energy used by a network of 1,625 nodes, with both geographic routing and outerspace geographic routing. We have done more experiments with different network sizes, up to 10,000 nodes, and the result does not change.

Usually, when a wireless network consumes more energy, its life is shorter. However, it is not always the case. Sometimes it is better to consume more energy, if this is done more equally in the network. This is exactly what happens with outer space geographic routing. We consider four measures of network longevity: time to first node death, time to loss of efficiency in message delivery, time to loss of network area coverage, and time to network disconnection. These measures are well-known, used in the literature [60, 61, 62], and collectively cover most of the concerns related to network lifetime. We have made four sets of experiments, each using one of the above ways to measure the longevity of the network. In each of the experiments we count the number of messages that are successfully delivered before network "death", where network death is defined according to each of the above four measures.

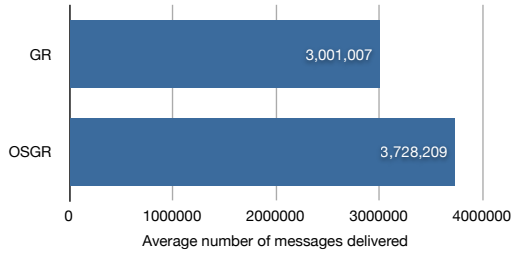
The first set of experiments is done according to the first measure. We have generated a



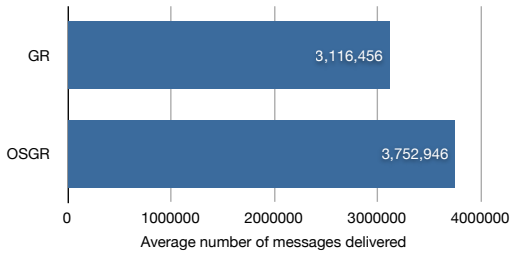
(a) Time to first node death.



(b) Time to loss of efficiency in message delivery.



(c) Time to loss of area coverage.



(d) Time to network disconnection.

Figure 1.5: The time is measured as the number of messages delivered to destination before the death of the first node. The network consists of $1,625$ nodes. GR stands for geographic routing while OSGR stands for outer space geographic routing.

network, a uniform traffic, and injected the traffic into two copies of the same network, one using geographic routing and one using outer space geographic routing. This have been iterated several times with networks of different sizes. The result is shown in Figure 1.5(a), where we show the number of messages delivered on average by a network of $1,625$ nodes (the result does not change by considering network of different size), using both routing protocols.

As you can see, the network lifetime when using outer space geographic routing is 29.17% longer on average than geographic routing. As a matter of fact, the number of messages successfully delivered by the network until the very first node death is much larger with routing in outer space. Figure 1.5(b) shows the result we get when considering the second definition of network lifetime. In this case, we consider the network dead when it is not efficient any more in delivering messages. Note that geographic routing (and similarly its outer space version) has the problem of "dead ends", places where the message

cannot proceed because there is no node closer to destination, while the destination is still far. There are a number of solutions to this problem (see for example [49]), and there do exists more sophisticated versions of geographic routing that know how to deliver a message whenever there is a path between source and destination. These solutions can be used both by geographic routing and by outer space geographic routing. However, when the network is not able any longer to deliver messages without these sophisticated add-ons, that means that the network is deteriorated. We use this as a measure of the quality of its structure. In this set of experiments we count the number of messages that reach destination until the success ratio of message delivery falls under some threshold (in our case 95%). As it can be seen in the figure, also in this case outer space geographic routing wins and prolongs the life of the network by 12.54% on average.

The third set of experiments is related to area coverage. One of the main application scenarios of sensor networks is the monitoring of some area of interest. In such applications, a must in terms of network properties is the fact that the area of interest has to be fully covered by the network sensing power. Of course, as long as the nodes begin to die, achieving this task becomes more and more difficult. We have performed our experiments assuming that sensing radius is 0.1, just like transmission radius. Again, outer space geographic routing is better and guarantees area coverage much longer. From Figure 1.5(c), you can see that routing in outer space increases network lifetime of 24.23% when considering coverage.

Lastly, the fourth set of experiments consider network lifetime until network disconnection. Note that connectivity is one of the most important network properties, and that it is *different* from network coverage. Also in this case, outer space geographic routing wins over geographic routing. As it can be seen from Figure 1.5(d), with routing in outer space the network lives 20.42% longer, on average.

Since security usually comes at a price, this is somewhat surprising. Routing in outer space delivers a network that, simultaneously, offers less front for an attack and is more energy-efficient according to several different definitions of network lifetime.

1.4 Applications of routing in outer space

1.4.1 Uniform distribution of nodes in the random way-point mobility model

The random way-point mobility model [13] is one of the most classical models in the literature of mobile computing. The model is simple: Each node moves independently in the network and iterates a procedure in which it chooses a way-point in the area uniformly at random, it moves straight to the way-point with uniform speed chosen at random, it waits a randomly chosen period of time at the way-point, and finally it iterates by choosing another way-point. The reader is surely able to see immediately why the distribution of the mobile nodes is not uniform in the network when this mobility model is used on the square for some time, even though they were deployed uniformly at time 0. Intuitively, the straight paths followed by the nodes from way-point to way-point are random segments on the square and therefore the nodes tend to concentrate at the center of the network—this is not different from the phenomenon that generates hot-spots when routing messages.

This problem is well-known. The stationary properties of the random way-point mobility model have been studied in [63, 64], among others. However, routing in outer space gives a clean and simple way to have uniform distribution of the nodes in the random way-point mobility model. When moving from one way-point to the other, do it by using the outer space shortest path! Immediately, this yields uniform distribution. Indeed, the random way-point mobility model on the torus does generate uniform distribution of the nodes (the torus is symmetric under any rotation; to see a formal argument about the same property on the sphere, see [64], the proof for the torus is the same). Note that this outer space version of the random way-point mobility model is *different* from random walk with reflection [64], which also has uniform distribution of the nodes but the node chooses a random direction instead of a random destination.

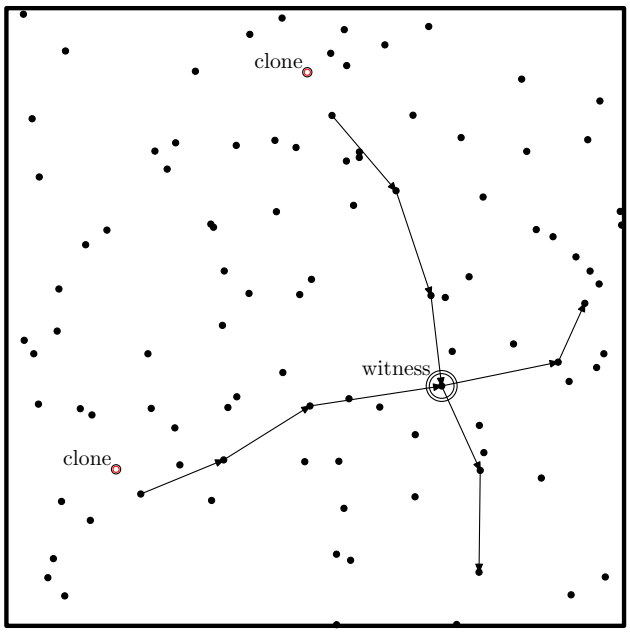
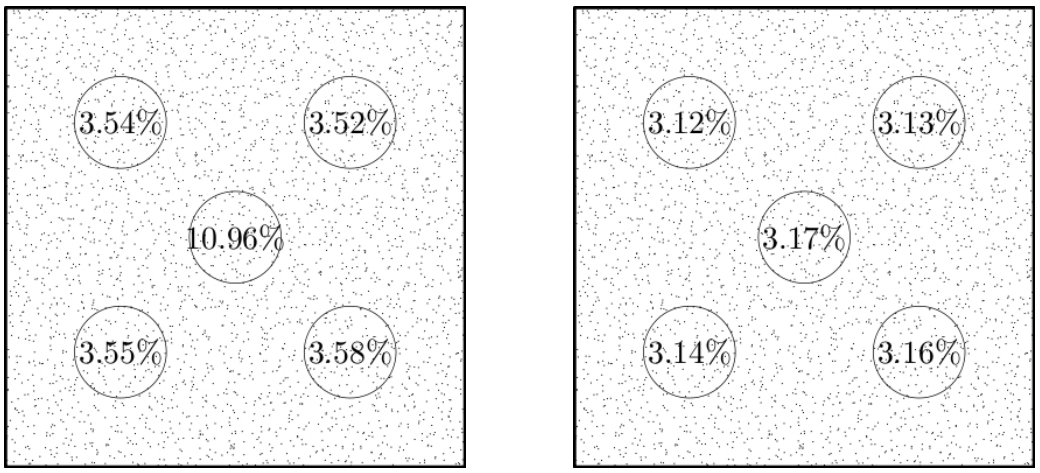


Figure 1.6: A run of LSM. Two clones of the same node are present in the network. In this example, two neighbors of the clones send the location claim to a random destination. The two paths intersect at a node, the witness, and the attack is thus detected.

1.4.2 Distributed detection of replication attacks in wireless sensor networks

Wireless sensor networks are often deployed outdoor or in hostile environments. In this cases, an adversary can physically capture some of the nodes, re-program and replicate the nodes in a large number of clones, and re-deploy them in the network. The clones are provided with the same cryptographic material as the originals, therefore they can fully communicate with the legitimate nodes and participate in the network operations such as data aggregation, consensus protocols, etc.. This gives the adversary the capability of launching all sorts of vicious insider attacks. The detection of the *node replication attack* is thus an important problem in wireless sensor networks and there has been a large amount of work on this topic [65, 66, 67, 68].

To the best of our knowledge, one of the most feasible and efficient solutions to this problem is the Line-Selected Multicast (LSM) protocol, a distributed approach introduced



(a) LSM with geographic routing.

(b) LSM with outer space geographic routing.

Figure 1.7: Witness distribution of LSM with geographic routing and outer space geographic routing.

in [67] that provides globally-aware, distributed node-replica detection. LSM is based on a routine that executes at fixed intervals of time. The routine works as follows: Every node announces its location to its neighbors with a signed claim; each neighbor locally checks both the signature and the location claimed in the message and forwards it with probability p to a fixed number $g \geq 1$ of randomly selected destinations. Each node on the path to destination checks the signature of the claim, locally stores the message and compares it with other location claims received during the same iteration of the detection protocol. If two clones are present in the network, there is a probability that some node receives two incoherent location claims—two claims with same node id and different location (see Figure 1.6).

When this is the case the node is a *witness* of a node replication attack and can trigger a revocation protocol for the node id. Iteration after iteration, the probability that a node replication attack goes undetected gets smaller and smaller, and tends to zero very quickly.

LSM is a simple, intuitive, and efficient protocol. Also, it generates a uniform traffic pattern, routed by using geographic routing [67]. As we know, such conditions give

rise to congested areas and therefore to all the previously mentioned security and energy-efficiency issues. More than that, as it is also observed in [68], the congestion phenomenon is exacerbated and even more pronounced if you consider the distribution of the witnesses—the intersection between the paths of two claim messages is much more likely to occur in a congested area of the network. This is also what we observed by simulating the LSM protocol over a square. Each individual experiment consists of one iteration of the basic routine of LSM. We run a very large number of experiments—one million of messages—and measured the percentage of witnesses in different sub-areas of the network. Figure 1.7(a) shows the average of the results on different networks of different size.

Almost 11% of the witnesses belong to an area in the center of the network whose size is only 3.14% of the network. The percentage decreases significantly in sub-central areas of the same size—the number of witnesses in the areas placed at the middle-half diagonals is about 70% lower. A "smart" adversary [68] can perform powerful attacks to this detection protocol in many ways. For example, it can jam the small area in the center (of size 3.14% of the network) stopping 11% of the possible witnesses. Or it can simply subvert nodes starting from this central area. In such a way LSM loses part of its efficiency and the probability of detecting clones decreases significantly. Clearly, these problems would not arise if the distribution of the witnesses were uniform.

One way to get uniform distribution of the witnesses is to run LSM on top of outer-space geographic routing. Indeed, outer space shortest paths are uniformly distributed and so path intersections are. We run the same set of experiments shown in Figure 1.7(a) with this idea and got the results shown in Figure 1.7(b). As predicted by our theory, the number of witnesses in each area is independent from its position and shows a virtually perfect distribution on the nodes of the network. This way we get improved strength against the attacks that we mentioned above. Moreover, it is interesting to see that also efficiency in detection is improved! Indeed, paths are longer and thus intersect with higher probability. In our multiple set of experiments, done with networks of different size (from 1,000 to 10,000 nodes), the probability of detection is 53% higher when using outer space geographic routing instead of geographic routing.

To summarize, routing in outer space guarantees that LSM have uniform distribution of the witnesses (and thus more strength against a few "smart" attacks), higher detection

efficiency, and longer life thanks to the improved energy efficiency of the routing layer.

1.5 Conclusions

Uniform traffic injected into multi-hop wireless networks and routed with shortest path like protocols generates congested areas. These areas carry a number of non-trivial issues regarding security, energy-efficiency, and tolerance to (a particular case of) selfish behavior. In this chapter we described routing in outer space, a mechanism to transform shortest path routing protocols into new protocols that do not have the above mentioned problems.

Routing in outer space guarantees that every node of the network is responsible for relaying the same number of messages, on expectation. We have shown that a network that uses routing in outer space does not have congested areas, does not have the associated security-related issues, does not encourage selfish positioning, and, in spite of using more energy globally, lives longer of the same network using the original routing protocol, according to a set of measures for network lifetime that collectively cover all the major concerns usually considered in the literature.

Lastly, routing in outer space has a few clean applications in mobility and security protocols. We have shown that a state of the art protocol for node replica detection like LSM [67] gets improved detection efficiency, more security against “smart attacks”, and more longevity just using outer space geographic routing instead of geographic routing, as done in [67], as the routing layer.

Chapter 2

Pocket Switched Networks And Human Mobility

Nowadays people walk around carrying all sorts of devices such as cellphones, PDAs, laptops, etc. Typically, these devices are able to communicate with each other in short distances by using communication technologies such as Blue-tooth or WiFi. The new network paradigm that rises from this intermittent communication, also known as Pocket Switched Networks (PSN [8, 14]), can be a key technology to provide innovative services to the users without the need of any fixed infrastructure. Pocket Switched Networks fall in the class of the Delay Tolerant Networks (DTNs) [69]. In DTNs, messages can multi-hop from source to destination by using the forwarding opportunities given by the contacts between the nodes. As its predecessor DTN, the PSN paradigm aims to provide end-to-end communication in mobile systems of intermittent connectivity, heterogeneous and social-based contact opportunities. The complexity of these systems brings new interesting challenges to the research community.

This chapter shortly introduces previous research in Pocket Switched Networks and reviews the main results that have served as insights in our work in this area.

2.1 Challenged networks and the quest for new approaches to forwarding in social mobile systems

During the last two decades wireless technology has become a part of our everyday life. We have seen wireless cellular systems develop to first, second and third generation in less than 20 years. In such systems the devices need access points to operate. The access points provide connectivity to devices in their proximity, and assist mobile users in keeping connected to the wireless system when they roam from one access point to the other. Without such fixed pre-existing infrastructure the users would not be able to access the system. Various ways of host and router mobility and the multitude of devices that network through wireless technology gives rise to challenged network architectures such as the Delay Tolerant Networking [69]. Examples of such networks include Terrestrial Mobile Networks, Exotic Media Networks, Military Ad-Hoc Networks, and Sensor and Sensor/Actuator Networks.

The architecture for Delay Tolerant Networking (DTN) [69] seeks to address the communication needs of these challenged environments where end-to-end communication paths not always exist. This architecture proposes a message based store-and-forward overlay network that leverages a set of convergence layers to adapt to a wide variety of underlying transports. In addition, the model also espouses novel approaches to application structuring and programming interface, fragmentation, reliability, and persistent state management.

The further development of the electronic and wireless technology has turned wireless devices into gadgets we carry around easily. These devices are becoming more and more powerful from the resources point of view. Think of the last generation cellphones, notebooks, PDAs etc. Not only are these devices are able to connect to wireless WiFi routers, they can also be networked through short range connection technologies such as Blue-tooth. The mixing of existing network infrastructure with ad-hoc mobile short range networking can turn the dream of “Internet available every time and everywhere” into reality. Another consideration to be made is that the communication between users is not always necessarily to pass through the Internet. According to a questionnaire survey among 70 participants in the Computer Laboratory University of Cambridge, around 50% of their email exchanges are among people they met daily [12]. Moreover, when we need some content is more

likely that we can easily find it in our friends' devices. After all, we usually share the same interests with them. Hence it would be useful to have a network paradigm that allows for cache searches between devices of individuals that are frequently in contact with each other.

The Pocket Switched Network (PSN) [8] is a new type of DTN that aims to provide all these services exploiting store-and-forward message strategy adapted to human mobility. In PSNs nodes are short range communicating devices carried by humans. Wireless communication links are created and dropped in time, depending on the physical distance of the device holders. Thus, human mobility creates communicating opportunities that define the very network. Examples of such network include people in working places, students in university campuses, and citizens in metropolitan areas.

The different degrees of social relationships between individuals, their personal and group interests, the communities to which they belong to etc. give just a flavor of the multitude of ingredients in human mobility, and thus, of the complexity of the PSN that it determines. Such complexity becomes a huge impediment to the end-to-end communication between network devices, the lack of which makes the PSN useless. State of the art routing algorithms for DTN networks [70, 71] still provide forwarding by building and updating routing tables whenever mobility occurs. Unfortunately this approach is not applicable to environments of unpredictable and rapid changing mobility such as the human movement. Rather, as the authors in [8] point out, forwarding in PSN can benefit from the use of emerging characteristics of the network that are tolerant to mobility, such as time lasting relationships among individuals in real life. It is thus natural to think that the first step to make in designing forwarding protocols for PSNs is to study human mobility and its social-based components in order to understand its properties and mobility independent structures that may emerge.

2.2 Properties of Human Mobility

With the aim to fully understand human mobility and its properties various research groups from all around the world have conducted different experiments with real wireless devices

distributed to people in different scenarios. In all experiments the data collected is related to the contact opportunities among network nodes. Various work has been done in analyzing these data sets in order to observe inherent statistical properties presented by human mobility in different environments. Metrics such as *inter-contact time* and *contact duration* obtained a major attention from the research community. The inter-contact time corresponds to how often people see each other, and thus, characterizes the opportunities of packet forwarding between nodes. Contact duration, which limits the duration of each meeting between people in mobile networks, limits the amount of data that can be transferred.

In [14] the authors are the first to analyze real data traces gathered from the experiments they make distributing wireless devices to people in conference, university and urban environments. The devices used in the experiments are Intel iMotes, a small platform designed for embedded operation, comprising an ARM processor, Blue-tooth radio, flash RAM, powered by CR2 battery. The iMotes were configured to perform a Blue-tooth baseband layer "inquiry" discovering the MAC addresses of other Blue-tooth nodes in range. The results of inquiry were written to flash RAM, recording contact periods between devices, in the form of {MAC, start time, end time}. We will refer to these datasets as *Cambridge 05*, *Cambridge 06*, *Infocom 05*, and *Infocom 06*.

- In *Cambridge 05* [72] the iMotes were distributed to students of the University of Cambridge and were programmed to log contacts of all visible mobile devices. The number of devices that were used for this experiment is 12. This data set covers 5 days.

- In *Cambridge 06* [73] the authors repeated the experiment using more devices. Also, a number of stationary nodes were deployed in various locations around the city of Cambridge UK. The data of the stationary iMotes will not be used in this chapter. The number of mobile devices used is 36 (plus 18 stationary devices). This data set covers 11 days.

- In *Infocom 05* [74] the same devices as in *Cambridge* were distributed to students attending the Infocom 2005 student workshop. The number of devices is 41. This experiment covers approximately 3 days.

Experimental data set	Cambridge 05	Cambridge 06	Infocom 05	Infocom 06
Device	iMote	iMote	iMote	iMote
Network type	Bluetooth	Bluetooth	Bluetooth	Bluetooth
Duration (days)	5	11	3	3
Granularity (sec)	120	600	120	120
Devices number	12	54 (36 mobile)	41	98 (80 mobile)
Internal contacts number	4,229	10,873	22,459	191,336
Average Contacts/pair/day	6.4	0.345	4.6	6.7

Table 2.1: The four experimental data sets

- In *Infocom 06* the previous experiment was repeated this time with more (80) participants.

Further details on the real traces are shown in Table 2.1.

The authors in [14] are the first to plot the CCDF (complementary cumulative distribution function) of inter-contact times between devices in these traces. They show empirically that such distribution follows a power law (of the form of Equation 2.2.1) over a wide range of values that span timescales of a few minutes to half a day.

$$p(x) \propto Cx^{-\alpha} \tag{2.2.1}$$

This empirical finding has motivated the authors to pose the hypothesis that inter-contact time has a CCDF with power law tail. Under this assumption, they derived some interesting results on the feasibility and performance of opportunistic forwarding algorithms. In particular, their hypothesis implies that for any forwarding scheme the mean packet delay is infinite, if the power law exponent of the inter-contact time is smaller than or equal to 1 (the case suggested to hold in practice by the empirical results so far). These results are in sharp contrast with previously known findings on similar packet forwarding algorithms (e.g. [75]) which were obtained under a hypothesis of exponentially decaying CCDF of inter-contact time. Later on, Karagiannis at al. in [76] by studying the same data traces find that the CCDF of inter-contact time between mobile devices features a dichotomy: It follows closely a power-law decay up to a characteristic time (that varies with the experiment)

beyond which, the decay is exponential. The dichotomy has important implications on the performance of opportunistic forwarding algorithms and implies that previous statements on performance of such algorithms may be over-pessimistic.

More recently, Barabasi et al. in [77] study the trajectory of a very large (100K) number of anonymized mobile phone users whose position is tracked for a six-months 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 are heavy tailed and also are different for different groups of users and similar inside each group. Furthermore, they plot also the frequency of visiting different locations and show that it is well approximated by a power law with an exponential decay.

In parallel with the study of the statistical properties of the human mobility Hui et al. in a list of successive works [8, 14, 78, 12] study the social aspect of PSNs, where people make the network. People are connected by relationship ties of different strength. These ties define a social network. In the case of PSNs where people carry the computer devices, the social network can be mapped to the computer network. Human mobility can be represented in the form of weighted graphs called *contact graphs*, where nodes represent devices and weighted arcs are derived from the frequency of contacts/contact duration among two nodes.

Exploring the properties and under covered sub-structures of the contact graphs gives important insights on the design of forwarding protocols for PSNs. The authors in [78] explore the presence of underlying community structures in real data traces. Qualitatively, a community is defined as a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network. The authors use the definition of the k–clique community given in [79]. A k-clique community is defined as a union of all k–cliques reachable from each other through a series of adjacent k–cliques. Two k-cliques are said to be adjacent if they share $k-1$ nodes. The authors use the k–clique community algorithm to undercover community structures of the contact graphs derived from the real traces. Since the real experiments have been made by the same authors of the

paper, they had an a-priori knowledge of the real social communities among participants to the experiment. They compare the real communities with the ones derived from the movement traces and observe a high similarity between the two. This finding reinforces the idea that human mobility is guided by social relationships among people in everyday life. Thus, social ties among individuals become an important tool in the design of adequate protocols for Pocket Switched Networks.

2.3 Human mobility and forwarding

As discussed earlier in this chapter the unpredictable nature of the human mobility makes end-to-end communication in PSN hard to obtain. To date, *Epidemic Forwarding* [9] remains the protocol with the highest performance in terms of *success delivery* and *delay*. The success delivery is defined as the percentage of the sent messages that are actually delivered to destination before the message timeout. The delay is measured as the average time of delivery of these messages. Epidemic forwarding is based on flooding: Upon contact between A and B, node A forwards message m to node B unless B already has a replica of m. As it can be imagined, this protocol is inefficient in terms of message replicas created in the network, and network bandwidth. Further work has been done in designing efficient and highly performing forwarding protocols for both DTNs and PSNs. Most of them utilize flooding techniques aiming to approximate the performance of Epidemic by trying to keep a low cost. *Spray and Wait* [80] limits the number of copies to a number specified a-priori and dependent on the environment.

Other forwarding techniques exploit the "probability of delivery" of nodes. Schemes based on this technique forward messages in a greedy fashion to nodes that are more likely to meet the destination. One of the most representative protocols in this group is *SimBet Routing* [11], where ego-centric centrality of nodes and their social similarity is used to compute the forwarding quality of a possible relay node. SimBet is very efficient in terms of message replicas created in the system, but also very well performing: It succeeds in delivering messages even when there is a unique time-path between nodes. Unfortunately this routing protocol needs for the nodes to maintain and update routing tables. This makes

it unfeasible in PSNs where the movement of the nodes is highly dynamic and very unpredictable.

Delegation Forwarding [10] is another protocol that falls under the same group of protocols of SimBet. In Delegation Forwarding, each node is given a forwarding quality that can vary from the overall number of contacts to the time of last contact, and from the contact frequency to the time of last contact with the destination. Similarly, each message is given a rate that initially matches the forwarding quality of the source node. Upon contact between A and B, every message m in A with a rate lower than the forwarding quality value of B updates its rate to this value. A forwards to B those messages whose rate is updated, unless B already has a replica. Intuitively, the rate of a replica indicates the value of the highest quality node that that replica has seen so far. Delegation is shown to have a very high performance, similar to Epidemic, yet keeping the cost much lower.

Finally, the last group of forwarding protocols for PSNs is based on techniques that exploit social aspects of node movement such as being part of a community. The most representative protocol of this group is *Bubble* [12]. This protocol bases its forwarding strategy on the popularity of the individuals within their communities and within the whole network. Messages are first forwarded to popular nodes within the network, till they reach the target community. Then, the forwarding continues within the community through the most popular nodes, until delivery to target nodes.

No matter what technique they use, all these protocols rely on the altruistic cooperation among nodes, which, in this settings where nodes are independent individuals cannot be given as granted. Therefore, the problem of building mechanism and protocols that can tolerate selfish behavior is an important and modern issue in the design of networking protocols for social mobile systems such as PSNs.

In the next chapter we study the impact of selfish behavior in Pocket Switched Networks. We start with the observation that even in the case of Epidemic, the forwarding performance decays significantly in presence of rational nodes. The phenomena is exacerbated in the case of more selective protocols such as Delegation. We then propose their selfish-tolerant alter egos: G2G Epidemic and G2G Delegation and show that both these

protocols are Nash Equilibria. Moreover we support our findings by a large set of experiments that also show that G2G Epidemic and G2G Delegation have the same performance as their vanilla alter egos yet lowering significantly the cost.

Chapter 3

Forwarding in Social Mobile Wireless Networks of Selfish Individuals

So far, all the forwarding protocols for PSNs, including the ones that cope with social aspects of human mobility, assume full cooperation and fairness among nodes. Selfish behavior of individuals is not considered, even though it is an inherent aspect of humans, the device holders.

In this chapter we explore the issue of forwarding in social mobile networks of selfish individuals, and show how even protocols such as Epidemic, that use multiple forwarding paths, break down immediately in presence of misbehaviors. The situation is even worse with more selective protocols, such as Delegation Forwarding. We address the problem, and present two protocols that exploit social aspects of the network in detecting unwanted selfish nodes. We assume that all the nodes are selfish and show formally that both protocols are Nash equilibria, that is, no individual has an interest to deviate. Extensive simulations with real traces show that our protocols introduce an extremely small overhead in terms of delay, while the techniques we introduce to force faithful behavior have the positive side-effect to reduce the number of message replicas present in the network. We test our protocols also in the presence of a natural variation of the notion of selfishness—nodes that are selfish with outsiders and faithful with people from the same community. Even in this case, our protocols are shown to be very efficient in detecting possible misbehavior.

The results presented in this chapter are included in [5].

3.1 Selfishness in Mobile Ad-Hoc Networks

Often, primitives for distributed systems such as gossiping, file sharing ecc. are based on the cooperation among nodes. These primitives suffer from the presence of selfish individuals whose aim is to maximize their benefit by limiting as much as possible their contribution to the system tasks. Thus the mitigation of selfish behavior in distributed systems has obtained considerable attention from the research community. See, as an important example, the work in [81, 82].

Mobile ad-hoc networks where devices have limited resources are even more affected by selfish behavior of nodes. Communication relies on multi-hop paths between source and destination, often resulting in bandwidth and battery consuming from the intermediate nodes' point of view. Selfish behavior in addition to intermittent connectivity turns the multi-hop network into into a one-hop communication architecture, which, as shown in [75], decreases dramatically the network capacity.

Previous work has been done in studying techniques of mitigation of selfish behavior for mobile ad-hoc networks. The solutions can be classified in two main approaches: *Reputation based schemes* and *credit based schemes*. In reputation based schemes, nodes collectively detect misbehaving members and propagate declaration of misbehaving throughout the network. Eventually this propagation leads to other nodes avoiding routes through selfish members. In credit based approaches nodes pay and get paid for providing service to others. Digital cash system is implemented in order to encourage correct behavior among nodes.

The work in [83] is the first to applicate reputation based techniques to selfish mitigation in in mobile ad-hoc networks. The authors propose a solution based on watchdogs and pathrates. They implement these tools on top of Dynamic Source Routing protocol (DSR) [84]. The watchdogs operate by listening to packet transmissions and detecting nodes that do not correctly forward to next hop relays. The pathrater exploits information gained from the watchdog in the choice of paths that are more likely to correctly forward packets to destination. The authors support their finding by validating their misbehaving detection technique using the Random Way Point mobility model (RWP) [13]. They observe empirically that their scheme increases the throughput of the network up to 60%.

This solution, though, does not punish selfish behavior: The avoidance of paths containing misbehaving nodes does not affect their benefit in receiving and sending messages. Rather, it lowers even more their costs in term of memory, bandwidth and battery consumption thus rewarding their selfishness.

In [85] the authors present the CONFIDANT protocol that aims to achieve exactly the opposite. The protocol also operates on top of DSR routing and it consists of four components: The monitor that acts as a watchdog and detects misbehaving in a nodes neighborhood; the trust manager that sends alarm signals to nodes alerting them about the presence of misbehaviors. This presence could either be detected by the node itself or it could be reported from another node. The reputation system is the third component of the CONFIDANT protocol and it manages a table where node ratings are stored and updated according to detected misbehaviors. When the rating of a node in the table exceeds a given threshold in addition to multiple forms of misbehaving, the fourth and last component of the protocol, the path manager, is activated. This component provides to re-rank paths according to ranking table changes, deletes paths containing misbehaving nodes and ignores route requests sent from them. The authors validate their scheme using the RWP model.

Again in [86] a reputation scheme that works on top of source routing protocols such as DSR is presented. The scheme is based on acknowledgment packets called TWOACK. When a packet is received the node sends a TWOACK (two-hops-acknowledgement) back through the active source route. If the source of the packet does not receive such acknowledgement by the two-hop-relay node it declares the next hop's forwarding link to be misbehaving and the respective route broken. The routing protocol then uses these claims to avoid broken routes and misbehaving links so increasing the network throughput. Again, the RWP mobility model is used to validate the scheme.

The second approach to mitigating selfish behavior in ad-hoc networks is the credit based technique. In [87] the authors propose a scheme based on digital cash system that encourages correct behavior of nodes. The nodes are initially provided with a a certain amount of digital currency called *nuggets*. Each time a node wants to use the system (for example to forward a packet) it has to pay the relay node in nuggets, otherwise no service is provided. The nodes are thus encouraged to forward other nodes' packets since this is the

only way to obtain additional nuggets in the system. Additionally, the nuggets system also prevents nodes from overloading the network with messages. The authors propose two rewarding schemes: In the *Packet Purse Model* the sender loads the packet with nuggets that serve to reward relaying nodes in the forwarding path. The amount of nuggets a relay is rewarded with depends on the distance of the forwarding hop (longer links correspond to more nuggets). In the *Packet Trade Model* relays first "buy" and then "sell" for more the packet by thus incrementing their nuggets. In this case the forwarding cost is covered by the destination node.

The problem with the Packet Purse Model is the difficulty in predicting the number of nuggets necessary to forward the packet to destination. If the prediction is excessive, the nodes will run out of nuggets very fast. On the other hand, if the prediction is lower than the necessary, the packet will be dropped as soon as the nuggets therein will end. The second approach prevents this problem but does not prevent a node from overloading the network. However each node is free to buy or refuse a packet by thus limiting such phenomena. Both approaches rely on the presence of a tamper-resistant piece of hardware called *security model* within nodes. The security model takes care of the correct maintaining of sensible information such as cryptographic material, nuggets counter etc.

In [88], a combination of the reputation based with the credit based scheme is presented. The protocol avoids the use of unified currency and introduces the concept of "justified selfishness" for nodes in non favorable topological location. This protocol though relies on a very strong assumption: at least one route composed of only well-behaved nodes must exist between any two well-behaved nodes.

In [89] the same authors of [87] present an improved scheme that addresses the problems raised from their previous approach: The difficulty in predicting the forwarding cost in nuggets of the packet in the Packet Purse Model, and the possibility of network overload in the Packet Trade Model. When a source node individuates the next rely of a packet it wants to send, the security models of both nodes establish a *secure association*. Information on the hosting nodes and on both security models are collected. Before the first node forwards the packet to the relay node, it has to pass it through its own security model. The expected number of nuggets needed to deliver it to destination is calculated and the number of the remaining nuggets is decreased accordingly. The sender's security model identifier

and other security information is added as a *security header* of the packet.

When a relay node B receives a packet from a node A, it passes it through its own security model. The security model of B checks whether the information within the security header is pertaining to the security association that he has of $A's$ security model within its own database. If this test passes then B accepts to forward the packet. For the rest, the relay node's security model adds its own security header to the packet and forwards it to the next relay node.

If A is not the source of the packet, B increases the *pending nuggets counter* of A by one. This is done for the purpose to postpone the increment of $A's$ nuggets number only after the next hop B correctly receives the packet. The security models run nuggets synchronization protocol that provides to transfer pending nuggets and resetting to 0 the pending nuggets counters. The authors also study the behavior of their mechanism through extensive simulations using the RWP mobility model.

Regardless of the performance of these schemes on ad-hoc networks, none of them is designed for social mobile networks such as PSNs. In addition, all previous protocols have been validated using the Random Way Point mobility model. As we will discuss further in the next chapter, this model is shown to be inadequate in simulating human mobility [90, 63, 15, 16].

Recently in [91] the authors introduce a barter-based cooperation system that aims to increase message delivery in opportunistic networks. The authors assume that altruistic static nodes scattered on the network area generate messages down-loadable from interested network members that pass by. When two nodes meet they exchange the list of the messages in their buffers and each node decides to download from the other node only from the subset of the messages to which it is interested. Then the nodes start downloading one message per node at time slot, till they move out each other's communication range. The game-theoretical model developed helps the authors prove that the approach foster cooperation among the nodes. They support their findings with extensive simulations done with the restricted random waypoint model RRW model and the Simulation of Urban MObility SUMO [92]. Though it introduces a novel technique of cooperation stimulation, their work is oriented to a gossip-like service within the network, where messages are created from

special nodes and have no specific destination. Moreover, the setting is different and the solution does not consider social aspects of pocket switched networks.

In [93], the authors study for the first time the impact of different distributions of altruism on the throughput and delay of mobile social communication system. They show that, when forwarding algorithms that use multiple paths are considered, social mobile networks are robust to different distributions of altruism of nodes. To the best of our knowledge their work is the first study aimed to explore altruistic/selfish behavior in PSNs, where mobility is guided by social relationships among individuals, and presents different levels of heterogeneity.

In this chapter, we introduce two forwarding protocols Give2Get Epidemic Forwarding and Give2Get Delegation Forwarding, which are, to the best of our knowledge, the first protocols for packet forwarding in Pocket Switched Networks that tolerate selfish behavior. We reach this goal by showing formally that no rational node has any incentive to deviate. In other words, our two protocols are Nash equilibria. We describe our methodology and the main steps, the mechanisms, and the idea that we have used to build the complete proof. Lastly, we perform a large set of experiments to check the performance of G2G Epidemic Forwarding and G2G Delegation Forwarding. Quite surprisingly, we discover that some of the mechanisms that we introduced to make these protocols Nash equilibria, are also useful to control the number of replicas in the network and push the messages quickly and cheaply far from the community where they have been generated. As a result, G2G Epidemic Forwarding and G2G Delegation Forwarding, besides providing robustness in a network where every node is selfish, have nearly the same delay and success rate of their original alter ego, and a considerably lower cost in terms of number of replicas (around 20% less).

3.2 The system model

In our system model, every node is selfish. This is a realistic scenario, if people can get the same level of service without consuming part of their battery or part of their wireless uptime or memory without any consequence, they will. And as soon as the first user finds a way to get more (or the same) by paying less, and publish the patch of the system software,

everybody will download the patch and use it. So, it is reasonable to assume that, if some of the nodes deviate selfishly, after a while everybody will.

We assume that there are no byzantine nodes in the network. We will also assume that selfish nodes do not collude. All the nodes in the system are interested in receiving and sending messages, in other words, all the nodes are interested in staying in the system. Nodes are loosely time synchronized. Loose time synchronization is very easy to get, if a precision in the order of the second is enough, like in our protocol. We assume that every control message of our protocols is labelled with a time-stamp, though it does not appear in the protocols to clean the presentation. The clock is used to check the timeouts, and the time-stamp is used when reporting to the authority a misbehavior.

Lastly, nodes are capable of making use of public key cryptography—this capability will be used to sign messages and to make sender to destination encryption. It is known that public key cryptography is more expensive than symmetric cryptography. However, modern cryptography techniques, like those based on elliptic curves, provide short signatures (a secure signature based on elliptic curves is just 160 bits long), and cheaper and cheaper computation [94], which is shown to be adequate even for sensors. Moreover, in our study we are addressing a network of smart-phones or PDAs, which are not-so-small devices. Modern smart-phones can run sophisticated applications, like decoders of streaming videos, 3D games, web browsers that can open SSL sessions, and others. For these devices, a signature per message can be considered a relatively low overhead. Therefore, we assume that every node has a public key and the corresponding private key. The public key is signed by an authority that is trusted by every node in the system. Note that the authority is never used actively in the protocols, except when a node has to be removed because it has not followed the protocol faithfully, which is an exceptional case.

In the rest of this chapter, we will use $H()$ to denote a hash function, and $\langle m \rangle_A$ to denote a message m signed by node A.

3.3 Give2Get Epidemic Forwarding

As discussed in Chapter 2, Epidemic Forwarding [9] uses every contact as an opportunity to forward messages. If node A meets node B, and A has a message that B does not have,

the message is relayed to node B. Epidemic forwarding is often used as a benchmark, it is easy to see that it is impossible to get smaller delay, or higher success rate. However, the overhead in terms of number of copies of the same message of the network is very high. Put simply, many of the forwarding protocols in the literature on Pocket Switched Networks have the goal of reducing drastically the overhead without affecting much the delay and the success rate of Epidemic Forwarding.

However, Epidemic Forwarding does not tolerate a scenario in which users can make selfish choices. Indeed, selfish nodes would simply drop every message they receive (except those destined to themselves!). In this section, we will show how to build a version of epidemic forwarding, called Give2Get Epidemic Forwarding, that works in a system in which every node is selfish. We will see that G2G Epidemic Forwarding is a Nash equilibrium, that is, no selfish node has a better choice than following the protocol truthfully. Most of the ideas and techniques that we develop in this section will be used in the more sophisticated protocols we introduce later in this chapter.

G2G Epidemic Forwarding consists of three phases: Message generation, relay, and test. Message generation executes when one node has a message to send to some other node in the system. Suppose that node S has a message to send to node D. The message is built according to the following form: $m = \langle D, E_{PK_D}(S, msg\_id, \text{body})\rangle_S$. Key PK_D is the public key of the destination D. Note that it is a precise design choice to hide the sender of the message to every possible relay except the destination. We will see later why it is important.

3.3.1 G2G Epidemic Forwarding: The relay phase

Once the message is generated, the sender S tries to relay it to the first *two* (at least) nodes it meets. Assume that node S meets node B. Node S starts a session with the possible relay by negotiating a cryptographic session key with node B. This is easily and locally done by using the certificates of the two nodes, signed by the same authority. In this way, both identities are authenticated. From this point on, every communication during the session is encrypted with a symmetric algorithm like AES and the session key (to keep the notation clean, this encryption is not shown in the protocols). Node S starts the relay phase by

$$A \xrightarrow{\langle \text{RELAY\_RQST}, H(m)\rangle_A} B \tag{3.3.1}$$

$$A \xleftarrow{\langle \text{RELAY\_OK}, H(m)\rangle_B} B \tag{3.3.2}$$

$$A \xrightarrow{\langle \text{RELAY}, H(m), E_k(m)\rangle_A} B \tag{3.3.3}$$

$$A \xleftarrow{PoR(m,A,B)\equiv\langle \text{POR}, H(m), A, B\rangle_B} B \tag{3.3.4}$$

$$A \xrightarrow{\langle \text{KEY}, H(m), k\rangle_A} B \tag{3.3.5}$$

Figure 3.1: Protocol of the relay phase (in case node B does not have the message).

asking node B if it has already handled a message with hash $H(m)$ (see Figure 3.1, where the role of S is described as done by node A) step 1). In case node B has never seen this message, the relay phase goes on (step 2), otherwise node B informs S that it should not be chosen as a relay. Note that node B would not lie, since it still does not know the content of the message, its destination, and, in particular, if node B itself is the destination. In other words, if B deviates and execute a modified version of the protocol in which it declines offers of being a relay without knowing the destination of the message, it won't receive any message, against its own interest. Node S generates a random key k, and sends message m to B, encrypted with key k (step 3). Then, node B sends a *proof of relay* to node S which in turn, lastly, sends key k to B, who now knows whether it is the destination of the message or just a relay.

3.3.2 G2G Epidemic Forwarding: The test phase

Node B, once it realizes that it is a relay for message m, will follow the same protocol as done by node S—find two other nodes and relay the message to these two nodes by executing the relay phase as shown in Figure 3.1. By doing so, it can collect two proofs of relay that it will be asked to show, when meeting node S again, during the test phase. Only when two proofs are collected the message can be discarded from B's memory. After a timeout Δ_1, B can stop looking for relays, and after an additional timeout Δ_2, node B can discard every information regarding the message. Timeout Δ_1 is chosen, just like in

$$A \xrightarrow{\langle \text{POR\_RQST}, H(m), s\rangle_A} B \tag{3.3.6}$$

$$A \xleftarrow{\langle \text{POR\_RESP}, PoR(m,B,X), PoR(m,B,Y)\rangle_B} B$$

or (3.3.7)

$$A \xleftarrow{\langle \text{STORED}, H(m), s, HMAC(m,s)\rangle_B} B$$

Figure 3.2: Protocol of the test phase.

Epidemic Forwarding, in such a way that the success rate is high enough, timeout Δ_2 is chosen in such a way that, with non-negligible probability, nodes B meets node S again before Δ_2 expires. Here, we are using the good properties of social networks, if S and B meets, then it is likely that they will in the near future. Our experiments in the following section fully support this hypothesis. Moreover, we will see that the delay of G2G Epidemic Forwarding is very close to the delay of Epidemic Forwarding, and that Δ_1 can thus be chosen as in its original alter ago without affecting the success rate. We will also see that it is safe and efficient to choose $\Delta_2 = \Delta_1$.

The test phase is started by node S (see Figure 3.2, where, again, the role of S is described as done by node A), when meeting node B, after timeout Δ_1 has expired. During the test phase, node S challenges node B: Either it has two proofs of relay, or it still stores the message. In case node B has two proofs of relay, it can replay with the two proofs. The challenge is a simple cryptographic protocol in which node S generates a random seed s and asks node B to send a keyed-Message Authentication Code $HMAC$ on message m. The particular HMAC used in this protocol should be designed in such a way to be heavy to compute, since we want to incentive node B to relay the message and get the two proofs of relay. Since B does not know the seed beforehand, it must be storing the message unless it has found two relays. Note that the test phase is started only by the source of the message, not by intermediate relays. This is very important to get a Nash equilibrium: only the sender has the interest of checking. As a positive side-effect, the heavy $HMAC$ is virtually never executed if no node deviates from the protocol—it is extremely unlikely that the first two relays are not able to find two other nodes that have never seen the message.

3.3.3 G2G Epidemic Forwarding is a Nash equilibrium

In a Nash equilibrium, no node has an incentive to *unilaterally* deviate from the protocol. Therefore, to prove formally that G2G Epidemic Forwarding is a Nash equilibrium, we have to show that every single step of the protocol is executed, and that it is executed truthfully. The proof is quite technical and long, we keep it reasonably short, going through the most important steps without hiding critical details.

One of the driving forces of this proof is that every node has the ultimate interest of being part of the system, that is receiving messages and sending messages that are eventually received by the destination. Therefore, the sender of a message, node S, will initiate the message generation phase and will execute the relay phase with the first two nodes met. In case it deviates, its own messages would not be delivered, which is against its own interest.

Let's consider node B in the relay phase. Once node B is asked to be a relay of a particular message m, it does not know who is the destination of m. If node B deviates from the protocol and declines to be a relay, it will never receive any message, since it will discover the destination of the message only at step 5. So, node B will send the messages in step 2 and 4. Let's now consider node A in the relay phase. Node A is either the source of the message, or an intermediate relay. If node A is the source of the message, it will initiate the phase, since it is interested that the message is received, and will send the messages in steps 1, 3, and 5, for the same reason. It is trickier to understand why node A follows the protocol even in case it is an intermediate relay. It will since it does not know the sender of the message, so, there is a non-negligible probability that it is one of the two first relays from the sender and that it will be asked to show two proofs or relay. Note that node A will realize whether it is one of the first two relays only after the timeout Δ_1 has expired. On the other hand, it is interested in getting rid of the message as soon as possible, since a message typically uses much more memory than the proofs of relays, and, lastly, it does not want to be in the position of performing the heavy *HMAC* in step 7 of the test phase. Note that, if node A simply discard the messages, node S will report to the authority that A is misbehaving. Node S can prove that A has taken the message (it can show $PoR(m,S,A)$), while node A cannot prove either that it still has the message or that he has relayed it. In this case, the authority would remove node A from the system. Lastly, node A will also send

the message in step 5. Indeed, the message is extremely short and, if the key is not sent, node B freezes the session with node A until it gets the key. In case A deviates to a protocol in which step 5 is not executed, it won't be able to send and to receive any message through node B and, after many frozen sessions, it won't be able to send or receive any message at all, against its own interest. Hence, we have got the following result.

Lemma 3.3.1. *A rational node will follow all the steps of the relay phase truthfully.*

Let's consider the test phase. As described, only the sender will start the phase, and it will otherwise the other nodes will start dropping its messages. On the other hand, every relay cannot tell whether the previous relay is the sender or not (the sender is encrypted in the message), so they cannot tell whether they will be tested or not. Since nodes do not take the risk of being removed from the network, they will get two proofs of relays or keep storing the message. Note that the sender won't reveal to be the sender before the timeout Δ_1. Our experiments show that the probability of being the first relay after the sender are high enough that, if a node deviates from the protocol, this is detected very quickly.

Lemma 3.3.2. *A rational node will follow all the steps of the test phase truthfully.*

To summarize, if a node deviates rationally from G2G Epidemic Forwarding, then, with non negligible probability, the deviation is detected and the node is removed from the network. Therefore, we get the following result.

Theorem 3.3.1. *G2G Epidemic Forwarding is a Nash equilibrium.*

3.4 G2G Epidemic Forwarding: Experiments with real traces on detecting deviations

In this section, we report on the results of some experiments with real data, with the goal of understanding what is the impact of selfish behavior in Epidemic Forwarding. Then, we turn to G2G Epidemic Forwarding and test how good is our protocol in detecting possible deviations. In the case of Epidemic Forwarding, we consider *message droppers*—nodes

that use the system to send and receive messages and that just drop every message they happen to relay. We will see that message droppers can make the performance of Epidemic Forwarding drop quickly, and we will also see that G2G Epidemic Forwarding detects this kind of deviation right away.

3.4.1 Selfishness and selfishness with outsiders

In a social environment, it is natural to consider two different ways of being selfish. The first is just selfishness—nodes that can deviate from the protocol with the goal of maximizing their personal interest. The second is *selfishness with outsiders*—nodes that can deviate form the protocol for their personal interest only when this does not damage people from the same community. This notion is natural since it comes from our personal experience, some people can tend to be truthful with those they care about, and selfish with outsiders. Formally, it is just vanilla selfishness with a different objective function. However, it is useful to define it as an independent notion. To implement selfishness with outsiders, we use the *k-clique* algorithm [79] (also used in [12]) for community detection on each data trace. Nodes that are selfish with outsiders deviate from the protocol only in sessions with nodes from other communities.

3.4.2 The data set

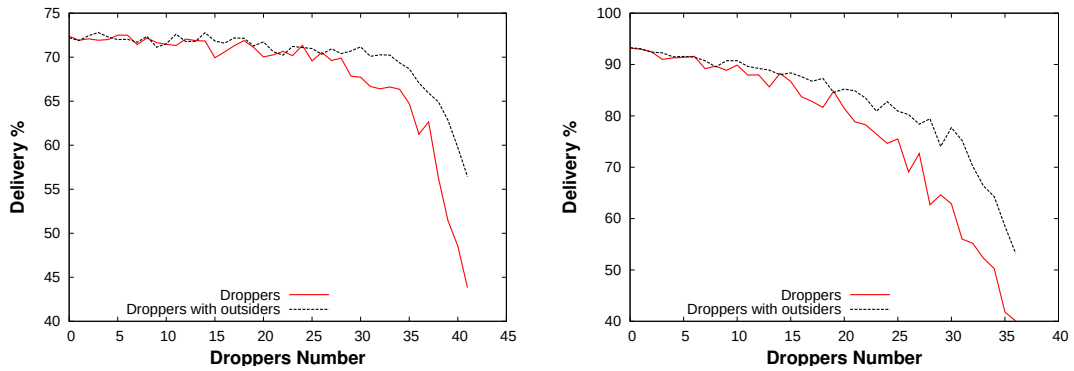
We use in our experiments two of the real traces presented in the previous chapter, *Infocom 05* and *Cambridge 06*. We excluded the dataset *Cambridge 05* because of the small number of the participating devices, while *Infocom 06* was not publicly available at the time of our study. Table 3.1 includes further details on these traces.

3.4.3 Impact of Selfish behavior on Epidemic Forwarding and detection of deviations in G2G Epidemic Forwarding

The assumptions and traffic pattern we use is as follows: A set of messages is generated with sources and destinations chosen uniformly at random, and generation times from a Poisson process averaging one message per 4 seconds. We isolated 3-hour periods for each

Experimental data set	Cambridge 06	Infocom 05
Device	iMote	iMote
Network type	Bluetooth	Bluetooth
Duration (days)	11	3
Granularity (sec)	600	120
Devices number	54 (36 mobile)	41
Internal contacts number	10,873	22,459
Average Contacts/pair/day	0.345	4.6

Table 3.1: The two experimental data sets



(a) Infocom05 (b) Cambridge06

Figure 3.3: Effect of message droppers on Epidemic Forwarding

data trace. Each simulation runs therefore 3 hours. To avoid end-effects no messages were generated in the last hour of each trace. The nodes are assumed to have infinite buffers.

We first focus on the effect of selfish behavior on Epidemic Forwarding. Figure 3.3 shows how success rate is affected by the presence of message droppers. As you can see, the performance of the protocol, under our assumption that every node is selfish, drops to around 50%, which is unacceptably low. Basically, when all the nodes are droppers, the only hope for success is that the sender gets personally in contact with the destination. It is not much different when we consider selfishness with outsiders. In the experiments we have used the k–clique community notion [79] discussed in the last chapter.

Next, we focus on the detection of message droppers. According to our experiments detection probability is more than 90%, both in the selfish case and in the selfish with

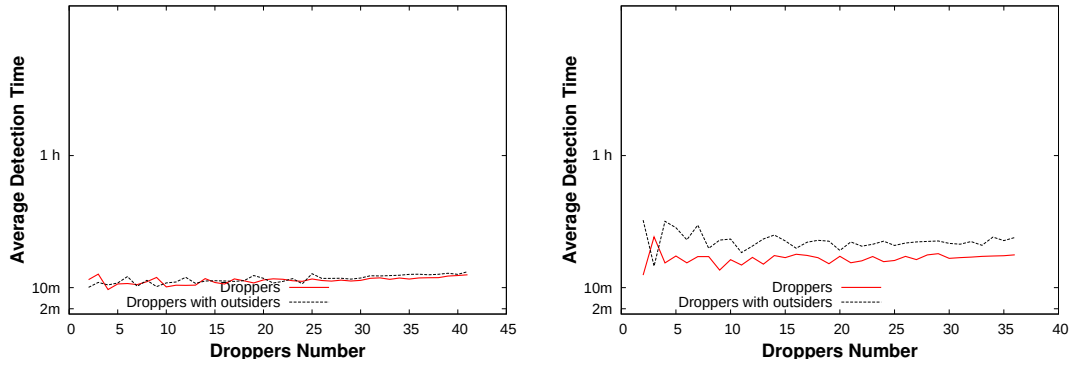
outsiders case. Deviations are detected very quickly, and the time does not depend on the number of the nodes that deviate (see Figure 3.4).

3.4.4 Selfish behavior in more selective forwarding protocols

Delegation Forwarding [10] is a class of protocols that have been shown to perform very well. As discussed in the last chapter, in Delegation Forwarding, every node is associated with a *forwarding quality*, that may depend on the destination of the message at stake. When a message is generated, it is associated with the forwarding quality of the sender. Then, the message is forwarded from node to node, creating a new replica of the message at each step, according to the following protocol. When a relay node A gets in contact with a possible further relay B, node A checks whether the forwarding quality of B is higher than the forwarding quality of the message. If this is case, node A creates a replica of the message, label both messages with the forwarding quality of node B, and forwards one of the two replicas to B. Otherwise, the message is not forwarded.

Delegation Forwarding, in many of its flavors, has been shown to reduce considerably the cost of forwarding (that is, the number of replicas), without reducing considerably success rate and delay. However, just like Epidemic Forwarding, it is far from being a Nash equilibrium. A selfish node can easily send messages and receive messages without taking care of relaying any other message. It is also easy to see that it is not enough to translate all the techniques used in G2G Epidemic Forwarding in order to get a version of Delegation Forwarding that is a Nash equilibrium.

Simply speaking, the techniques we developed to build G2G Epidemic Forwarding can prevent message dropping by those who take the message. However, selfish nodes has many other rational ways to deviate in these more sophisticated protocols. First, nodes can lie on their forwarding quality. They can claim that their quality is zero, and nobody can do much about this, these nodes would get their messages served without participating actively. We will call these nodes *liars*. Not only that, selfish nodes can change the forwarding quality of the message to zero, in such a way to get rid of the message soon—they would be able to relay it to the first two nodes they meet. We will call these nodes *cheaters*. Of course, cheaters are less vicious than liars, in our setting. However, we will show how to build a



(a) Infocom05 (b) Cambridge06

Figure 3.4: Detection of message droppers in G2G Epidemic Forwarding

version of Delegation Forwarding that is a Nash equilibrium. Just like what we did with G2G Epidemic Forwarding, our approach is not to add patches against liars and cheaters or incentives for altruistic nodes, our approach is to design a protocol such that, step by step, it can formally be shown that every rational player in the protocol cannot but following the protocol truthfully. In this way, we protect our system against liars, cheaters, and any other possible way to deviate rationally.

3.5 Give2Get Delegation Forwarding

In our study we consider Delegation Destination Frequency and Delegation Destination Last Contact [10].

Destination Frequency Node A forwards message m to node B if node B has contacted $m's$ destination more frequently than any other node that the copy of the message m carried by A has seen so far.

Destination Last Contact Node A forwards message m to node B if node B has contacted $m's$ destination more recently than any other node that the copy of the message m carried by A has seen so far.

G2G Delegation Forwarding builds upon all the techniques that we have developed for G2G Epidemic Forwarding. First, in G2G Delegation Forwarding the messages are

changed their quality only when forwarded (we will see later why). G2G Delegation Forwarding consists of four phases: Message generation, relay, test by the sender, and test by the destination. We will describe only the phases that are substantially different from G2G Epidemic Forwarding. Message generation is just like message generation and in G2G Epidemic Forwarding. In the following sections we show the key elements, without hiding important details.

3.5.1 G2G Delegation Forwarding: The relay phase and the test by the destination phase

Figure 3.5 shows the protocol of the relay phase. Just like G2G Epidemic Forwarding, node A has an interest to start this phase, since it has to collect the proof of relay for the message. In step 8, node A asks B what is its forwarding quality to D. Node B replies with its forwarding quality (we will see later why B has no interest in lying). When the destination of m is different from B, D is the actual destination; when the destination of m is B, D is a random node different from B. This mechanism has the goal of making it impossible to B to know whether it is the destination of the message or not before taking the message and giving the proof of relay. Therefore, just like in G2G Epidemic Forwarding, node B will follow all the relay protocol with the hope of receiving a message to itself. Note that in G2G Delegation Forwarding the proof of relay contains much more information, including the forwarding quality towards D claimed by node B and the forwarding quality of the message at that point in time.

Node A forwards the message to two other nodes. In the case when node A is also the sender of the message, A stores the signed message $\langle \text{FQ\_RESP}, B, D, f_{BD} \rangle_B$ for the nodes B that failed to be good relays for the message, that is $f_{BD} < f_m$. As soon as node A finds a good relay, the last two signed qualities of such failed relays are embedded into the message towards D. If the destination D receives the message, it will be able to check if f_{BD} is correct or not (this is the *test by destination* phase). Indeed, f_{BD} should be equal to f_{DB}. Since nodes B and C does not know whether A is the sender or not, they will not lie about their forwarding quality since there is a non-negligible probability to be tested by the destination (in the experiments we will see that this is exactly the case). When D detects

$$A \xrightarrow{\langle \text{FQ\_RQST}, H(m), D\rangle_A} B \tag{3.5.1}$$

$$A \xleftarrow{\langle \text{FQ\_RESP}, B, D, f_{BD}\rangle_B} B \tag{3.5.2}$$

$$A \xrightarrow{\langle \text{RELAY}, H(m), f_m, E_k(m)\rangle_A} B \tag{3.5.3}$$

$$A \xleftarrow{\langle \text{POR}, H(m), A, B, D, f_m, f_{BD}\rangle_B} B \tag{3.5.4}$$

$$A \xrightarrow{\langle \text{KEY}, H(m), k\rangle_A} B \tag{3.5.5}$$

Figure 3.5: G2G Delegation Forwarding: Protocol of the relay phase.

that node B is a liar, D sends a request of removal of B from the network to the authority. Note that, in our setting, we don't really need to introduce mechanisms to make this proof checkable by the authority, node D has no interest in lying. However, simple techniques can be introduced to make it impossible to D remove faithful nodes. For example, in case of Delegation Destination Last Contact, if the nodes exchange a signed message (with a time-stamp, as usual) at every contact, this message would be a proof of misbehaving against B. Similar techniques can be introduce for Delegation Destination Frequency.

In order to make this mechanism work, the forwarding quality f_{BD} is not the *current* quality, it is the quality computed *in the last completed timeframe*. Every node keeps three versions of the forwarding quality, the current and the two forwarding qualities computed in the previous two completed timeframes. In this way, B and D has a consistent notion of forwarding quality. (Of course, the timeframe has to be set in such a way that, with high probability, the message delay is much smaller.) As a consequence of this set of techniques, no relay will lie about their forwarding quality—we will see in the experiments that, in case of deviation, the probability of being removed from the system is actually very high.

3.5.2 G2G Delegation Forwarding: The test by the sender phase

The test by the sender is executed only by the sender of the message. Assume that node A is not the sender, and that it has received the message from the sender S. When A gets in contact with S again, after timeout Δ_1 (defined as in G2G Epidemic Forwarding), node A

is tested and, just like in G2G Epidemic Forwarding, it gives the two required proofs $\langle \text{POR}, H(m), A, B, D, f_m^1, f_{BD} \rangle_B$ and $\langle \text{POR}, H(m), A, C, D, f_m^2, f_{CD} \rangle_B$ to node S. In this way, it is guaranteed that it is not rational to became a message dropper. More than that, this phase is also important to check that A is not a cheater, that is it has not reduced f_m to get rid of the message quickly. Indeed, S can check whether

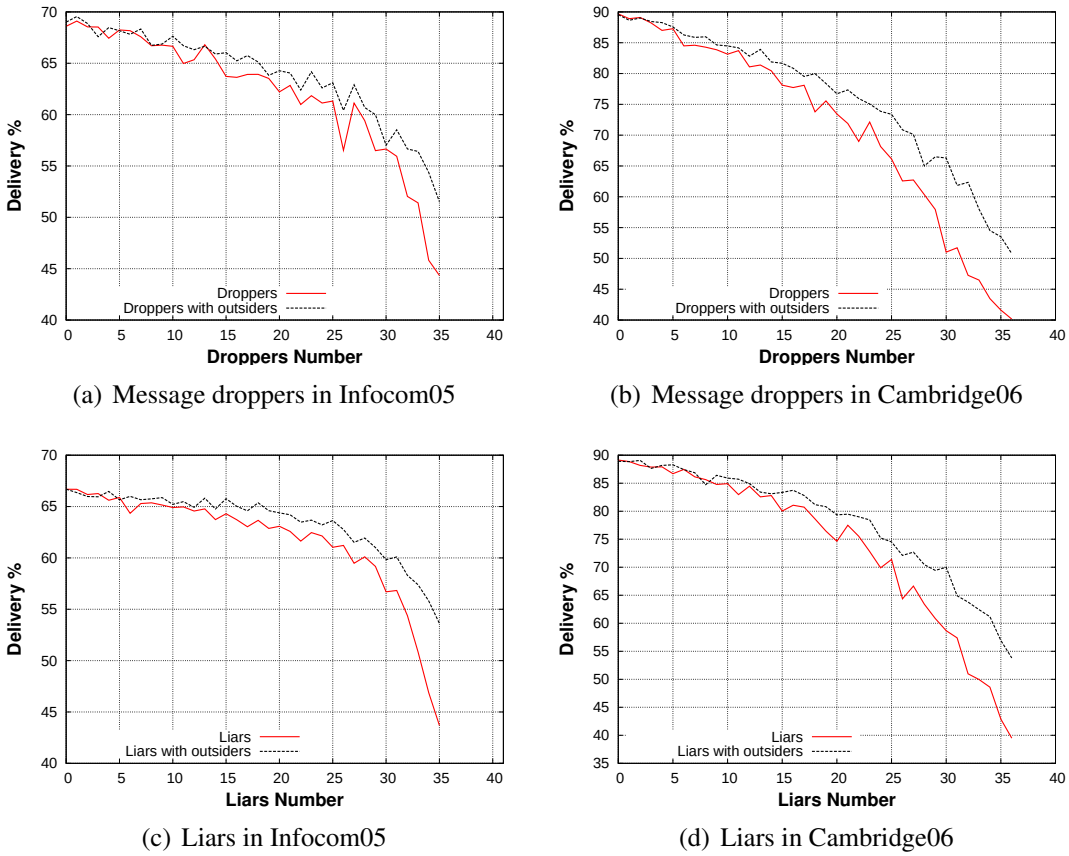
$$f_{AD} = f_m^1 < f_{BD} = f_m^2 < f_{CD}.$$

The second equality in this equation is true since the quality of the messages is changed only when forwarded. Since we know that nodes do not lie, than we know that f_{AD}, f_{BD}, and f_{CD} are sound. Therefore, also $f_m^1 = f_{AD}$ and consequently node A has not selfishly modified the forwarding quality of the message to convince B to take it. Similarly, we also know that $f_m^2 = f_{BD}$ and so node A has not cheated with node C as well. To summarize, we get the following result.

Theorem 3.5.1. G2G Delegation Forwarding *is a Nash equilibrium.*

3.6 G2G Delegation Forwarding: Experiments with real traces on detecting deviations

In this section we will consider message droppers, liars, and cheaters. Note that it is not rational to be a cheater in vanilla Delegation Forwarding—if a node labels the message with forwarding quality zero, than it will have to relay it with higher probability, doing more work. Since we are interested only in *rational* deviations, we will see what is the impact of droppers and liars for Delegation Forwarding, and how fast and reliably G2G Delegation Forwarding is able to detect droppers, liars, and cheaters as well. Recall that *no* deviation is rational in G2G Delegation Forwarding—it is a Nash equilibrium—and these experiments on the detection probability are important just to make sure that our assumption that every node has a non-negligible probability of being tested during the test phase is sound. Here we present the results for Delegation Destination Last Contact. Delegation Destination Frequency, as far as detection of deviations is concerned, behaves in a very similar way.



(a) Message droppers in Infocom05

(b) Message droppers in Cambridge06

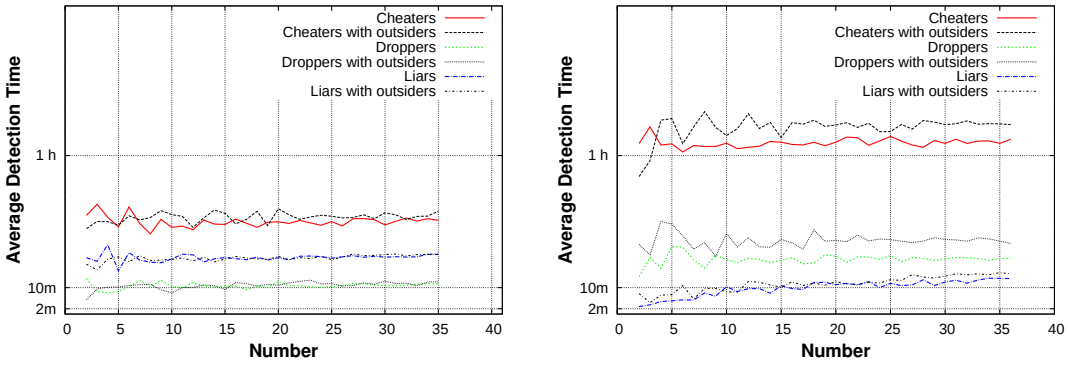
(c) Liars in Infocom05

(d) Liars in Cambridge06

Figure 3.6: Effect of message droppers and liars on Delegation Forwarding

We run a first set of experiments to see what is the impact of these deviations on Delegation Forwarding. Figure 3.6 shows the results, that clearly indicate that both droppers and liars have a big impact on the success rate, both in the case of selfishness and in the case of selfishness with outsiders.

Second, we have run a large set of experiments to see how reliably these deviations are detected by the protocols in both traces, Infocom 05 and Cambridge 06. According to our results, droppers and cheaters are detected with probability larger than 80% and liars with probability larger than 60%. In all cases, this is much more than enough to say that the probability of being detected is not negligible. Recall that, in our model, users have the interest of being part of the system, and that they are not willing to risk (even with small probability) to be removed from the network.



(a) Infocom05 (b) Cambridge06

Figure 3.7: Detection of message droppers in G2G Epidemic Forwarding

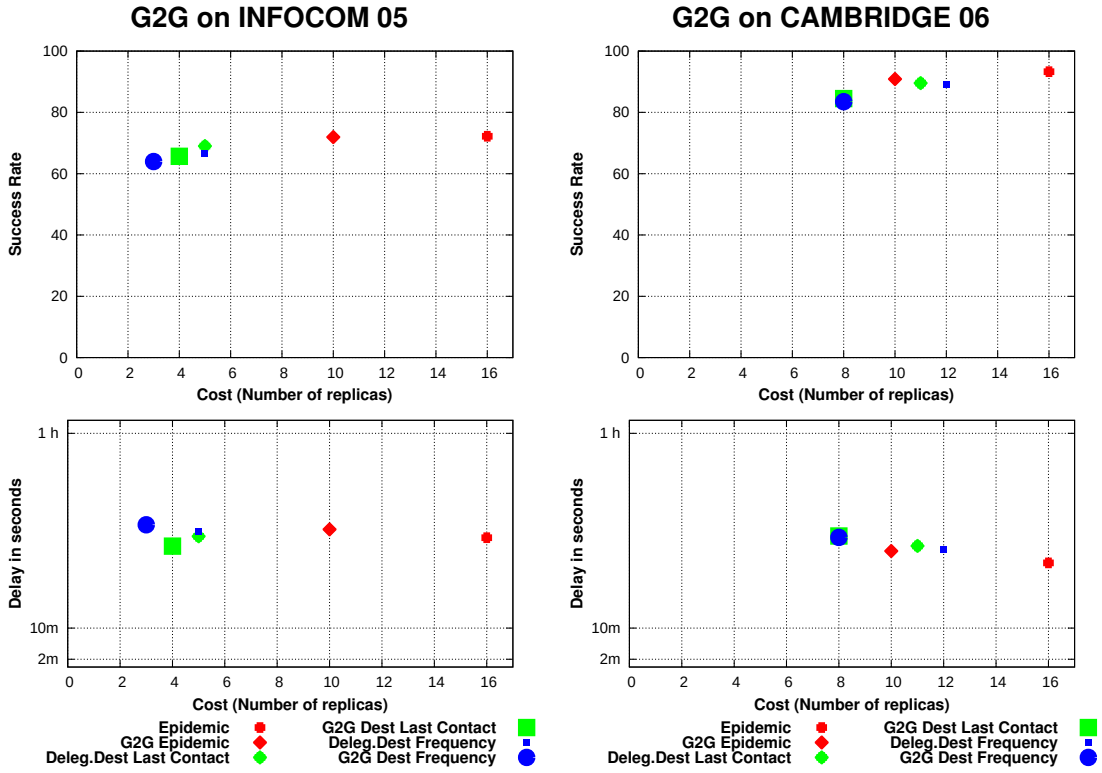
Then, our question is how fast is the detection. Figure 3.7 shows that node cannot hope to deviate and remain in the system for long time. The results indicate that G2G Delegation Forwarding is very fast in the detection of all the selfish behaviors considered. The case of selfishness with outsiders is detected not as fast, but still very quickly. Note that the time needed is longer in Cambridge 06, which has less frequent contact rate.

3.7 Experiments on the performance of G2G Epidemic Forwarding and G2G Delegation Forwarding

In this section we are interested in evaluating the performance of G2G Epidemic Forwarding and G2G Delegation Forwarding compared with their original alter egos. The experimental setting is the same that has been used throughout the whole chapter, and described in Section 3.4. We are interested in the following metrics: *memory* (amount of memory overhead of the protocol), *success rate*, *delay*, and *cost* in terms of number of replicas of the same message in the network. We start from memory. Indeed, we can easily check that the memory used by the G2G version of these protocols is within a constant factor from their original counterpart. It is enough to go through the protocols step by step.

Initially, a reasonable goal was to show that adding all the mechanisms and functionalities needed to make the protocols Nash equilibria does not reduce the performance considerably. During the protocols design, we realized that we could hope for more. The mechanism used to reduce the number of relays to two, besides being fundamental to show that the protocol is a Nash equilibrium, has the interesting property that the message more cheaply flows far from the community that generated it. Cheaply in the sense that fewer replicas need to be generated to reach destinations that are far from the sender in terms of community. Figure 3.8 summarizes a long set of experiments on success rate, delay, and cost. Indeed, looking at the results, something that might seem surprising is happening—G2G Epidemic Forwarding is much better than Epidemic Forwarding in terms of cost, and G2G Delegation Forwarding is considerably better than Delegation Forwarding, again in terms of cost. Note that the experimental setting that we have chosen is considered to be standard in the literature.

To summarize the results of our experiments, G2G protocols show an excellent performance in terms of cost, even compared with their alter egos that are not Nash equilibrium, decreasing considerably (more than 20%) the number of replicas generated in the system, while their performance in terms of delay and success rate are very close to the original protocols.



(a) Infocom05 (b) Cambridge06

Figure 3.8: Performance of G2G Epidemic Forwarding and G2G Delegation Forwarding compared with Epidemic Forwarding and Delegation Forwarding

3.8 Conclusions

Pocket Switched Networks raise from mobile humans that carry short range communicating devices. The intermittent connectivity of these networks makes end-to-end communication difficult to obtain. The design of forwarding protocols for PSN has therefore obtained much attention from the research community. Different techniques have been proposed that rely on cooperation among nodes, totally ignoring the possible presence of selfish individuals in the network.

We test two well known forwarding protocols Epidemic and Delegation, in presence of misbehaving individuals, and observe a drastic decay of their performance. Then we present G2G Epidemic Forwarding and G2G Delegation Forwarding, the first protocols for message forwarding that work under the assumption that all the nodes in the network are

selfish. The two protocols cope with the social properties of the PSN networks and have very high detection rate. We test our protocols also in the presence of a natural variation of the notion of selfishness–nodes that are selfish with outsiders and faithful with people from the same community. Even in this case, our protocols are shown to be very efficient in detecting possible misbehavior We formally show that the G2G protocols are Nash equilibria. Quite surprisingly, G2G protocols also outperforms their alter egos in terms of cost, while being almost as good in terms of success rate and delay.

Chapter 4

Small World In Motion

Social Mobile ad-hoc networking has presented many challenges to the research community, especially in designing suitable, efficient, and well performing protocols. The practical analysis and validation of such protocols often depends on synthetic data, generated by some mobility model. The model has the goal of simulating real life scenarios [95] that can be used to tune networking protocols and to evaluate their performance.

In this chapter we present small world in motion (SWIM), a new mobility model for social mobile ad-hoc networking. 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, which we believe to be the base of human mobility in real life. SWIM is the first model to show experimentally and theoretically the presence of the power law and exponential dichotomy of inter-contact time distribution. We validate our model using real traces and compare the distribution of inter-contact time, contact duration and number of contact distributions between nodes, showing that synthetic data that we generate match very well real data traces. Furthermore, we show that SWIM can predict well the performance of forwarding protocols. We compare the performance of two well known forwarding protocols that we have discussed previously in this dissertation—Epidemic Forwarding [9] and (a simplified version of) Delegation Forwarding [10]—on both real traces and synthetic traces generated with SWIM. The performance of the two

protocols on the synthetic traces accurately approximates their performance on real traces, supporting the claim that SWIM is an excellent model for human mobility.

The results presented in this chapter appear in [4].

4.1 The quest for mobility models in PSN

Mobile ad-hoc networks are characterized by the movement of devices over an area of interest: Sensors in battlefield, satellites around a planet, cars provided with PDAs in a highway, humans in a city etc. When validating protocols for mobile systems, either it is possible to generate real case-scenarios, or a model of simulation has to be used. Though the former it is sometimes a feasible possibility, most of times the evaluation relies on the properties of some underlying mobility model. This is done for different reasons: Lack of researching funds, difficulty of creating some of the scenarios (e.g. underwater scenarios), lack of scalability etc. The model in use should be able to evaluate accurately the protocol, in the sense that, its performance on synthetic and on real data should be as similar as possible.

Till a few years ago, the model of choice in academic research in mobile ad-hoc networks was the random way point mobility model (RWP) [13], simple and very efficient to use in simulations. In this stochastic model each node moves independently, choosing its next destination (*waypoint*) as a random point over a squared network area. Then the node moves towards the waypoint with constant speed, picked randomly over an interval $[v_{min}, v_{max}]$. When the destination is reached, the node pauses for a time interval also chosen randomly over an interval $[t_{min}, t_{max}]$. Then this process is repeated again.

Being the RWP a benchmark in the evaluation of protocols for mobile ad-hoc networks, many researchers have focused their works on studying its properties from different points of view. In [90] for example, the authors prove that the average nodal speed in the RWP decreases over time. In simulations where $v_{min} = 0$, this value quickly decreases to zero, by thus turning the system in quasi-static one. The authors also show how the performance of several routing protocols for ad-hoc networks differs when evaluated in RWP with different simulation time. Later on, in [63] the authors prove that the distribution of the nodes over the network area in RWP changes in time. Even if the initial distribution is uniform, the

more the simulation goes on, the more the nodes tend to concentrate at the center of the network.

These two works gave insights on the inadequacy of RWP in evaluating protocols for mobile ad-hoc networks. The rise of the PSN networks brought even more proofs on the matter. As discussed in Chapter 2, with the aim of understanding human mobility many researchers have performed real-life experiments by distributing wireless devices to people. From the data gathered during the experiments, they have observed the typical distribution of metrics such as inter-contact time (time interval between two successive contacts of the same people) and contact duration. In [8, 14], the authors show that the distribution of inter-contact time is a power-law. Later, in [15], it has been observed that the distribution of inter-contact time is best described as a power law in a first interval on the time scale (12 hours, in the experiments under analysis), then truncated by an exponential cut-off. Conversely, [16] proves that RWP yields exponential inter-contact time distribution. Their work clearly established that models based on random movement like RWP are inadequate to simulate human mobility, and thus, they're not reliable in validating protocols for networks such as PSNs.

The inadequacy of the RWP mobility model raised the need of new, more realistic mobility models for social mobile ad-hoc networking. Moreover, with the public availability of the real data traces on the CRAWDAD database [96], the researchers focused on creating mobility models that generate synthetic data that match well the statistical properties of the real ones. In [97] the authors present a mobility model based on Levy Walks. The model is similar to a random walk, except that the flight lengths (movement distances) and the pause times in destinations are generated with power-law distribution. In the past, Levy Walks have been shown to approximate well the movements of animals. The movement produces inter-contact time distributions similar to real world traces. However, since every node moves independently, the model does not capture any social behavior among nodes.

In [98], the authors present a mobility model based on social network theory. The model takes in input a social network with clearly specified well established relationships among nodes. The authors discuss the community patterns and groups distribution in geographical terms. They validate their synthetic data with real traces and show a good matching

between them.

The work in [99] 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 [100], the authors present a mobility model that simulates the every day life of people that go to their work-places in the morning, spend their day at work and go back to their homes at evenings. Each one of this scenarios is a simulation per se. The synthetic data they generate match well the distribution of inter-contact time and contact durations of real traces.

In a very recent work, Barabasi et al. [77] study the trajectory of a very large (100,000) number of anonymized mobile phone users whose position is tracked for a six-months 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 are heavy tailed and also are different for different groups of users and similar inside each group. Furthermore, they plot also 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, as we will see, support the intuition behind our model SWIM.

4.2 SWIM: From intuition to real traces

We believe that a good mobility model should

1. be simple; and

2. predict well the performance of networking protocols on real mobile networks.

We can't overestimate 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 inter-contact 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. Lastly, and most importantly, a model should be 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 a real network. As accurately as possible.

None of the mobility models in the literature meets all of these properties. The random way-point mobility model is simple, but its traces do not look real at all (and has a few other problems). Some of the other protocols we reviewed in the related work section can indeed produce traces that look real, at least with respect to some of the possible metrics, but are far from being simple. And, as far as we know, no model has been shown to predict real world performance of protocols accurately.

We propose *small world in motion* (SWIM), a very simple mobility model that meets all of the above requirements. Our model is based on a couple of simple rules that are enough to make the typical properties of real traces emerge, just naturally. We will also show that this model can predict the performance of networking protocols on real mobile networks extremely well.

4.2.1 Nearby Restaurant or VIP Bar? The intuition behind SWIM

When deciding where to move, humans usually trade-off. The best supermarket or the most popular restaurant that are also not far from where they live, for example. It is unlikely (though not impossible) that we go to a place that is far from home, or 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, etc. These are the basic intuitions SWIM is built upon. Of course, trade-offs 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—trading-off 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.

4.2.2 The model in details

More in detail, to each node is assigned a so called *home*, which is a randomly and uniformly chosen point over the network area. Then, the node itself assigns to each possible destination a *weight* that grows with the popularity of the place and decreases with the distance from home. The weight represents the probability for the node to chose that place as its next destination.

At the beginning, no node has been anywhere. Therefore, nodes do not know how popular destinations are. The number of other nodes seen in each destination is zero and this information is updated each time a node reaches a destination. Since the domain is continuous, we divided the network area into many small contiguous cells that represent possible destinations. Each cell has a squared area, and its size depends on the transmitting range of the nodes. Once a node reaches a cell, it should be able to communicate with every other node that is in the same cell at the same time. Hence, the size of the cell is such that its diagonal is equal to the transmitting radius of the nodes. Based on this, each node can easily build a *map* of the network area, and can also calculate the weight for each cell in the map. These information will be used to determine the next destination: The node chooses its cell destination randomly and proportionally with its weight, whereas the exact destination point (remind that the network area is continuous) is taken uniformly at random over the cell's area. Note that, according to our experiments, it is not really necessary that the node has a *full* map of the domain. It can remember just the most popular cells it has visited and assume that everywhere else there is nobody (until, by chance, it chooses one of these places as destination and learn that they are indeed popular). The general properties of SWIM holds as well.

Once a node has chosen its next destination, it starts moving towards it following a straight line and with a speed that is proportional to the distance between the starting point and the destination. To keep things simple, in the simulator the node chooses as its speed

value exactly the distance between these two points. The speed remains constant till the node reaches 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 since 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.

More specifically, let A be one of the nodes and h_A its home. Let also C be one of the possible destination cells. We will denote with $seen(C)$ the number of nodes that node A encountered in C the last time it reached C. As we already mentioned, this number is 0 at the beginning of the simulation and it is updated each time node A reaches a destination in cell C. Since h_A is a point, whereas C is a cell, when calculating the distance of C from its home h_A, node A refers to the center of the cell's area. In our case, being the cell a square, its center is the mid diagonal point. The weight that node A assigns to cell C is as follows:

$$w(C) = \alpha \cdot distance(h_A, C) + (1 - \alpha) \cdot seen(C). \tag{4.2.1}$$

where $distance(h_A, C)$ is a function that decays as a power law as the distance between node A and cell C increases.

In the above equation α is a constant in $[0;1]$. Since the weight that a node assigns to a place represents the probability that the node chooses it as its next destination, the value of α has a strong effect on the node's decisions—the larger is α, the more the node will tend to go to places near its home. The smaller is α, the more the node will tend to go to "popular" places. Both small and big values for α rise clustering effect of the nodes. In the first case, the clustering effect is based on the neighborhood locality of the nodes, and is more related to a social type: Nodes that "live" near each other should tend to frequent the same places, and therefore tend to be "friends". In the second case, instead, the clustering effect should raise as a consequence of the popularity of the places.

When reaching destination the node decides how long to remain there. One of the key observations is that in real life a person usually stays for a long time only in a few places, whereas there are many places where he spends a short period of time. Therefore, the distribution of the waiting time should follow a power law. However, this is in contrast

with the experimental evidence that inter-contact time has an exponential cut-off, and with the intuition that, in many practical scenarios, we won't spend more than a few hours standing at the same place (our goal is to model day time mobility). So, SWIM uses an upper bounded power law distribution for waiting time, that is, a truncated power law. Experimentally, this seems to be the correct choice.

4.2.3 Power law and exponential decay dichotomy

In a recent work [15], it is observed that the distribution of inter-contact time in real life experiments shows a so called dichotomy: First a power law until a certain point in time, then an exponential cut-off. In [16], the authors suggest that the exponential cut-off is due to the bounded domain where nodes move. In SWIM, inter-contact time distribution shows exactly the same dichotomy. More than that, our experiments show that, if the model is properly tuned, the distribution is strikingly similar to that of real life experiments.

We show here, with a mathematically rigorous proof, that the distribution of inter-contact time of nodes in SWIM has an exponential tail. 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 *not* only depends on its own previous decisions, but also depends on *other nodes'* decisions: Where a node goes now, strongly affects where it will choose to go in the future, and, it will affect also where other nodes will chose to go in the future. So, in SWIM there are no renewal intervals, decisions influence future decisions of other nodes, 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 inter-contact time of nodes A and B is defined as:

$$T_I = \inf_{t>0}\{t : ||A(t) - B(t)|| \leq r\}$$

Assumption 4.2.1. *For all nodes A and for all cells C, the distance function distance*(A,C) *returns at least* $\mu > 0$.

Theorem 4.2.1. *If* $\alpha > 0$ *and under Assumption 4.2.1,* the tail *of the inter-contact time distribution between nodes A and B in SWIM has an exponential decay.*

Proof. To prove the presence of the exponential cut-off, 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 way point decision in the interval between t_i and t_{i+1} and that each node has enough time to finish a leg. Recall that this is of course possible since waiting time at way points is bounded above and since 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

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

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

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

Consider $\mathbb{P}\{\varepsilon_i | \varepsilon_{i-1} \cdots \varepsilon_1\}$. At time $t_i + \delta_i$, at least one of the two nodes is at a way point, 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 way point). During its last way point decision, node A has chosen cell C as its next way point with probability at least $\alpha\mu > 0$, thanks to Assumption 4.2.1. 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

Experimental data set	Cambridge 05	Cambridge 06	Infocom 05
Device	iMote	iMote	iMote
Network type	Bluetooth	Bluetooth	Bluetooth
Duration (days)	5	11	3
Granularity (sec)	120	600	120
Devices number	12	54 (36 mobile)	41
Internal contacts number	4,229	10,873	22,459
Average Contacts/pair/day	6.4	0.345	4.6

Table 4.1: The three experimental data sets

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,

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

So,

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

for sufficiently large t. □

4.3 SWIM vs Real traces

In order to show the accuracy of SWIM in simulating real life scenarios, we will compare SWIM with three data sets gathered during experiments done with real devices carried by people. More specifically, we will use the sets *Infocom 05*, *Cambridge 05* and *Cambridge 06*. We described how these traces have been collected in Chapter 2, therefore here we only recall basic details in Table 4.1.

Characteristics of these data sets such as inter-contact and contact distribution have been observed in several previous works [8, 101, 14]. Especially in [15] it has been empirically shown that the inter-contact time decays as a power law, followed by an exponential decay. In this section we explore empirically the inter-contact, contact duration and number of contacts between nodes in SWIM's traces, generated simulating each of the real scenario considered, and confront the results. But first, let us start with the simulation environment.

4.3.1 The simulation environment

For the evaluation of SWIM, we built a discrete even simulator of the model. The simulator is written in c++ programming language and takes 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 Equation 4.2.1;
- 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:

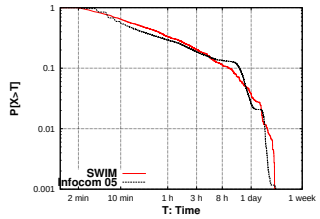
- *Meet* event: When two nodes are in range with each other. The output line contains the ids of the two nodes involved and the time of occurrence.
- *Depart* event: When two nodes that were in range of each other are not anymore. The output line contains the ids of the two nodes involved and the time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

In the output, we don't really need information on the geographical position of the nodes when the event occurs. However, it is just straightforward to extend the format of the output file to include this information. In this form, the output file contains enough information to compute correctly inter-contact intervals, number of contacts, duration of contacts, and to implement state of the art forwarding protocols.

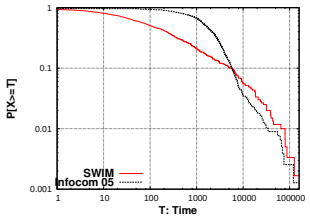
During the simulation, each sensor keeps a vector $seen(C)$ and updates it continuously. Note that the nodes do not necessarily agree on what is the popularity of each cell. As mentioned earlier, it is not necessary to keep in memory the whole vector, without changing the qualitative behavior of the mobile system. However, the three scenarios Infocom 05, Cambridge 05, and Cambridge 06 are not large enough to cause any real memory problem. Vector $seen(C)$ is updated at each *Finish* and *Start* event, and is not changed during movements.

4.3.2 The experimental results

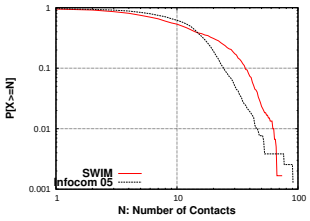
In this section we present some experimental results in order to show that SWIM is a simple and good way to generate synthetic traces with the same statistical properties of real life mobile scenarios. The idea is to tune the few parameters used by SWIM in order to simulate Infocom 05, Cambridge 05, and Cambridge 06. For each of the experiments we consider the following metrics: inter-contact time CCD function, contact distribution per pair of nodes, and number of contacts per pair of nodes. The inter-contact time distribution is important in mobile networking since it characterizes the frequency with which information can be transferred between people in real life. It has been widely studied for real traces in a large number of previous papers [8, 14, 101, 16, 15, 98, 102]. The contact distribution per pair of nodes and the number of contacts per pair of nodes are also important. Indeed they represent a way to measure relationship between people. As it was also discussed in [103, 104, 12] it's natural to think that if a couple of people spend more time together and meet each other frequently 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 social mobile environments such as PSNs [12].



(a) Distribution of the inter-contact time in Infocom 05 and in SWIM



(b) Distribution of the contact duration for each pair of nodes in Infocom 05 and in SWIM



(c) Distribution of the number of contacts for each pair of nodes in Infocom 05 and in SWIM

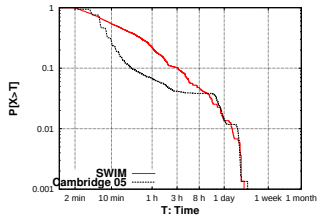
Figure 4.1: SWIM and Infocom 05

Let's now present the experimental results obtained with SWIM when simulating each of the real scenarios of data sets. Since the scenarios we consider use iMotes, we model our network node according to iMotes properties (outdoor range 30m). We initially distribute the nodes over a network area of size 300×300 m^2. In the following, we assume for the sake of simplicity that the network area is a square of side 1, and that the node transmission range is 0.1. In all the three experiments we use a power law with slope $a = 1.45$ in order to generate waiting time values of nodes when arriving to destination, with an upper bound of 4 hours. We use as $seen(C)$ function the fraction of the nodes seen in cell C, and as $distance(x,C)$ the following

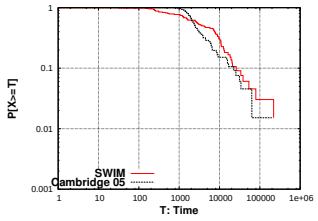
$$distance(x,C) = \frac{1}{(1+k||x-y||)^2},$$

where x is the position of the home of the current node, and y is the position of the center of cell C. Positions are coordinates in the square of size 1. Constant k is a scaling factor, set to 0.05, which accounts for the small size of the experiment area. Note that function $distance(x,C)$ decays as a power law. We come up with this choice after a large set of experiments, and the choice is heavily influenced by scaling factors.

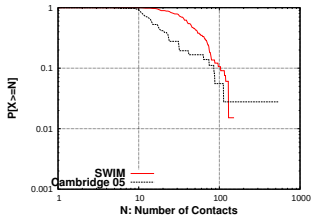
We start with the scenario of the Infocom 05 experiment. The number of nodes n and the simulation time are the same as in the real data set, hence 41 and 3 days respectively. Since the area of the real experiment was quite small (a large hotel), we deem that $300 \times$



(a) Distribution of the inter-contact time in Cambridge 05 and in SWIM



(b) Distribution of the contact duration for each pair of nodes in Cambridge 05 and in SWIM



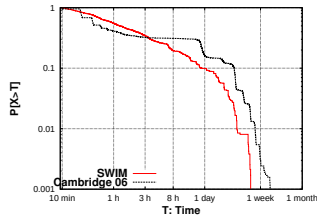
(c) Distribution of the number of contacts for each pair of nodes in Cambridge 05 and in SWIM

Figure 4.2: SWIM and Cambridge 05

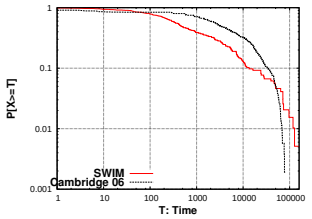
300 m^2 can be a good approximation of the real scenario. In Infocom 05, there were many parallel sessions. Typically, in such a case one chooses to follow what is more interesting to him. Hence, people with the same interests are more likely to meet each other. In this experiment, the parameter α such that the output fit best the real traces is $\alpha = 0.75$. The results of this experiment are shown in Figure 4.1.

We continue with the Cambridge scenario. The number of nodes and the simulation time are the same as in the real data set, hence 11 and 5 days respectively. In the Cambridge data set, the iMotes were distributed to two groups of students, mainly undergrad year 1 and 2, and also to some PhD and Master students. Obviously, students of the same year are more likely to see each other more often. In this case, the parameter α which best fits the real traces is $\alpha = 0.95$. This choice proves to be fine for both Cambridge 05 and Cambridge 06. The results of this experiment are shown in Figure 4.2 and 4.3.

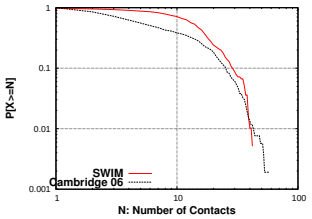
In all of the three experiments, SWIM proves to be an excellent way to generate synthetic traces that approximate real traces. It is particularly interesting that the same choice of parameters gets goods results for all the metrics under consideration at the same time.



(a) Distribution of the inter-contact time in Cambridge 06 and in SWIM



(b) Distribution of the contact duration for each pair of nodes in Cambridge 06 and in SWIM



(c) Distribution of the number of contacts for each pair of nodes in Cambridge 06 and in SWIM

Figure 4.3: SWIM and Cambridge 06

4.4 Comparative performance of forwarding protocols

The performance of protocols strictly depends on the underlying model used in their evaluations. So far we have seen how SWIM is able to generate traces that have similar statistical properties with the ones obtained from real life experiments. Now our goal is to check SWIM's accuracy in the evaluation of protocols. For this we use two well known forwarding protocols, already discussed in this dissertation: Epidemic Forwarding [9] and a simplified version of Delegation Forwarding [10] in which each node has a random constant as its quality. Of course, this simplified version of delegation forwarding is not very interesting and surely non particularly efficient. However, we use it just as a worst case benchmark against epidemic forwarding, with the understanding that our goal is just to validate the quality of SWIM, and not the quality of the forwarding protocol.

It's important to stress that for these experiments SWIM has been tuned in the same exact way as 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 have been used to fit real traces with synthetic traces.

The forwarding assumptions and the traffic generator is the same as in Chapter 3. 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 per 4 seconds. The nodes are assumed to have infinite buffers and carry all message replicas they receive until the end of the simulation. Again, as in Chapter 3, the

metrics we are concerned with are: *cost* (the number of replicas per generated message), *success rate* (the fraction of generated messages for which at least one replica is delivered), *average delay* (the average duration per delivered message from its generation time to the first arrival of one of its replicas). 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.

Here below we recall once again the function of the two forwarding protocols.

Upon contact with node A, node B decides which message from its message queue to forward in the following way:

Epidemic Forwarding: Node A forwards message m to node B unless B already has a replica of m. This protocol achieves the best possible performance, so it yields upper bounds on success rate and average delay. However, it does also have a high cost.

(Simplified) Delegation Forwarding: To each node is initially given a quality, distributed uniformly in $(0;1]$. To each message is given a rate, which, in every instant corresponds to the quality of the node with the best quality that message have seen so far. When generated the message inherits the rate from the node that generates it (that would be the sender for that message). Node A forwards message m to node B if the quality of node B is greater than the rate of the copy of m that A holds. If m is forwarded to B, both nodes A and B update the rate of their copy of m to the quality of B.

Figure 4.4 shows how the two forwarding protocols perform in both real and synthetic traces, generated with SWIM. As you can see, the results are excellent—SWIM predicts very accurately the performance of both protocols. 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 and important statistical properties like inter-contact time, number of contacts, and duration of contacts. Then, you can have a good confidence that the model is properly tuned and usable to get meaningful estimation of the performance of a forwarding protocol for PSN networks.

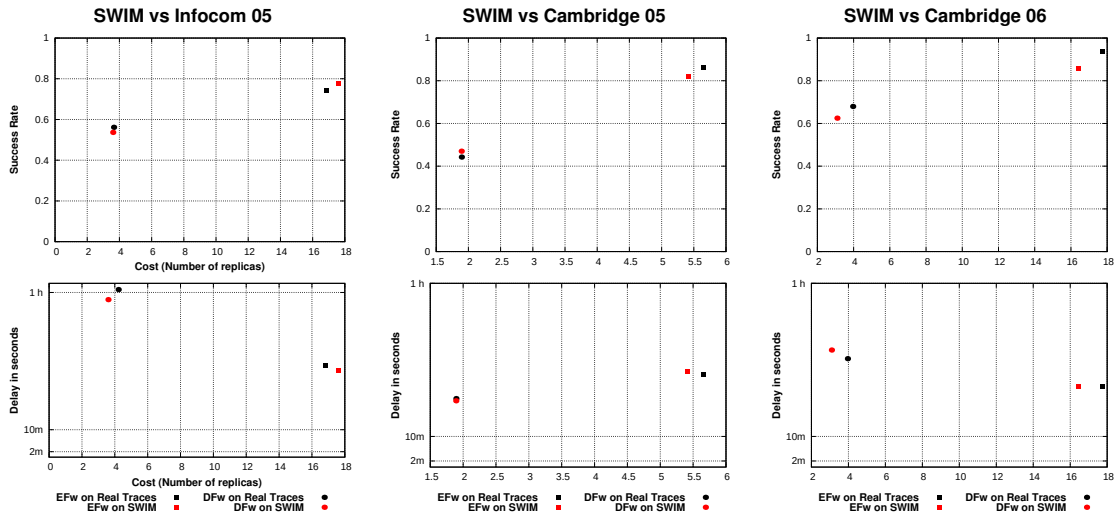


Figure 4.4: Performance of both forwarding protocols on real traces and SWIM traces. EFw denotes Epidemic Forwarding while DFwd Delegation Forwarding.

4.5 Conclusions

Protocol validation in social mobile networks is a difficult task. Real life experiments are hard to accomplish and expensive to realize, and often lack of scalability. The remaining possibility is simulating the reality through an underlying mobility model. Base requirements for a good model are simplicity, efficiency in simulations and accuracy in predicting performance of protocols. Random Way Point mobility model (RWP), considered till a few years ago as a benchmark from the research community, generates traces with an exponential decay of inter-contact times between nodes [16]. This result is in contradiction with recent findings [15] that show empirically a power-law followed by an exponential cut-off distribution of inter-contact times in real traces. Moreover, in RWP nodes lack of an important aspect of human behavior, the sociality, which guides human mobility and generates complex sub-structures in the PSN network that it defines.

In this chapter we presented SWIM, a new mobility model for social mobile wireless networks. SWIM movement is based on simple observations on everyday life: We prefer nearby, well-known places to faraway, unpopular ones; and the time spent in a few of them is considerably longer in comparison to our permanence in many many others. Nodes in SWIM chose their next destination trading off between popularity and distance from their

home point of the place. SWIM is the first model to show experimentally and theoretically the presence of the power-law and exponential dichotomy of inter-contact time distribution. It generates simulated scenarios that match the real ones in terms of inter-contact time, contact duration and number of contacts between nodes. It shows very good accuracy in performance evaluation of forwarding protocols for PSNs.

Chapter 5

Interest and Community Forwarding in Individual MObile NETworks

In the last three chapters of this dissertation we introduced the concept of wireless social mobile network as a new emerging communication paradigm. We reported on the vast literature on the topic, especially in exploiting knowledge of social ties between network members to optimize performance of traditional communication techniques such as forwarding. Then we discussed the need of considering the *individual* behaviors of network *idividuals*, such as selfishness, in the design of apt protocols for these networks.

In this chapter we want to push ourselves even further, putting the network individual and his needs in the center of our study. For this purpose we introduce the concept of INdividual MObile NETwork (IMONET), where each individual is associated with a *community profile* that characterizes him with a certain set of habits, interests and social relations within the network. We then use this characterization in the design of two novel communication paradigms: *Interest* and *community* casting. In interest-casting, a message characterized by a certain *relevance profile* is selectively forwarded to potentially interested IMONETs members, while in community-casting a message is selectively forwarded to the IMONET nodes who are members of a certain community. We then present the complete design of an interest- and community-casting protocol called CIF, and show through extensive simulations that CIF is able to disseminate information to interested IMONET

members almost as quickly as Epidemic Forwarding, but with a much smaller communication overhead.

The results presented in this chapter are included in [6].

5.1 Individual Mobile Network: a new prospective towards the future

The vision of a near future in which a multitude of hand-held devices establish direct wireless communication links in an opportunistic fashion has motivated researchers to focus their efforts on the PSNs [8], in which nodes are *individuals* carrying such powerful hand-held devices. As discussed earlier in this dissertation, the social component of the mobility pattern can be a very useful tool in designing apt protocols in this area. Indeed, the idea of exploiting information regarding social ties between network nodes in PSNs is not new. For instance, in [11] the authors propose using the notions of "ego-centric betweenness" and "social similarity" to improve end-to-end routing performance. In [12], the authors propose the use of a social "centrality" metric to achieve the same purpose. In [105], the authors use a "social similarity" metric locally computed from the history of past encounters to route messages within the network. Recently, a social-based approach based on a notion of "ego-centric betweenness" has been proposed also to optimize multicast performance [106].

The above approaches have shown how the social structure underlying a PSN can be successfully exploited to improve communication performance. However, existing approaches share the following drawbacks:

- *lack of scalability:* in order to characterize strength of social ties, a node must keep traces of *all* (or most) network nodes encountered in a very long time window; if the network is very large, keeping trace of all encountered nodes might impose a prohibitive storage overhead, especially considering that network nodes are typically resource-limited. The situation is even worse if these traces have to be exchanged between nodes (like, e.g., in [11, 105]).

- *global knowledge required*: in some cases, global knowledge about the community

structure in the entire network is needed. This is the case, for instance, of approaches based on the notion of centrality (e.g., [12]).

– *missed communication opportunities*: since existing approaches heavily rely on tracing past encounters, when two nodes meet for the first time they typically do not exchange messages. Hence, several communication opportunities are missed (think about how many individuals a certain person meet only once in a typical working day, for instance).

– *focus on mobility pattern*: individuals within a PSN are not considered as such – i.e., a person with a collection of interests, habits, social relationships, etc. –, but characterized based solely on their mobility pattern – i.e., based on the frequency and duration of contact with other members of the network. The focus on individual's mobility pattern, while very useful for optimizing performance of traditional networking paradigms (e.g., unicast), is not apt for realizing innovative networking paradigms for PSNs, specifically centered around an individual's interests, communities, etc.

– *communication paradigms typical of traditional networks*: the focus so far has been on optimizing performance of *traditional networking paradigms* such as unicast (e.g., [11, 12, 105]) and multicast (e.g., [106]). On the contrary, it is our strong belief that PSNs will manifest their full potential when innovative communication paradigms specifically designed for such networks will be realized. It is not clear why, for instance, a PSN member should use the opportunistic network to send a unicast message to another specific member, especially considering that incurred delays would be very long (in the order of hours or days), while alternative communication means are typically available (SMS, cell phone call, etc.). A similar observation applies to the multicast communication paradigm.

In order to address the above limitations, we advocate a different perspective on how information related to the user social behavior is used to optimize network performance. In particular, our aim is to regard network nodes as *individuals*, and not as *mobile tokens*. We use the term token here to highlight the fact that current literature mostly focuses on

mobility patterns, regardless of whom is actually holding the mobile device (that in principle might be installed on a mobile robot, hanged on the roof of a vehicle, etc.). Mobility patterns are then used to characterize what the literature call "social network" structure, which is indeed a "mobility pattern characterization": The fact that two network nodes are into each other reach does not necessarily mean that there is a social contact between them (think about two strangers sitting in the same metro carriage). On the contrary, in our approach we put the network user in the center of our study. We introduce the concept of *INdividual MObile NETwork* (IMONET for short), where the individual is characterized with a *community profile* describing his set of habits, interests and social relations within the network. Similarly, we associate messages circulating in the network with *relevance profiles*. We then use the notion of similarity between community and relevance profiles to drive information propagation within the IMONET. In particular, the proposed approach is:

- *stateless*, hence *scalable*: since information propagation is based only on comparison of community and interest profiles, which are then discarded, only minimal storage capability is required on the nodes. Storage capability is in principle independent of the number of nodes within the network.

- *no* or *very limited global knowledge* is required.

- *no missed communication opportunity*: information propagation is driven by the notion of similarity between community and interest profiles; hence, even individuals encountering for the first time might potentially exchange messages.

- *focus on individuals*, instead of mobility pattern. As we elaborate in the next section, this shift of focus we advocate facilitates the creation of innovative communication paradigms specifically designed for IMONETs.

The line of research closer to our work is the design of social-based publish-subscribe mechanisms for IMONETs. For instance, the authors of [107] considers a network in which a service provider (e.g., a cell phone operator) selectively sends dynamic content updates to users, updates that can be shared with other users when a communication opportunity

arises. The authors show that performance can be considerably improved if the relative frequency of updates sent to users from the service provider takes into account the strength of social ties between network users. In [108], the authors approach the problem of sharing data within an IMONET from a social-based perspective, with the goal of optimizing content availability through careful, social-aware data placement.

The work that is most closely related to ours is [109], in which the authors present the design of a mechanism whose underlying routing framework, called SocialCast, exploits predictions based on metrics of social interactions to drive the forwarding process. SocialCast uses social knowledge in a predictive way, allowing in principle message exchange even between individuals meeting for the first time. However, implicit in SocialCast is the assumption that an individual implicitly or explicitly subscribes to one or more "interests". On the contrary, our approach builds on the notion of *community profile*, in which an IMONET member compactly encodes not only the *degree* (not necessarily binary) of interest in different topics, but also his/her habits (e.g., where he/she lives, works, etc.). Thus, our approach allows a more complete characterization of an IMONET member's habits and social relationships. Finally, SocialCast still remains a publish-subscribe scheme and requires storing of a considerable amount of state information at the nodes, which should be contrasted with the stateless approach taken herein.

5.2 Interest- and Community-based networking

IMONETs should be used to create innovative services that can be effectively realized within the IMONET itself, without the need of resorting to pre-existing communication facilities. An example might be an *interest-casting* primitive, in which an IMONET user A wants to communicate a certain information (for instance, a film about Puccini opera composer displayed at a local theater) to the maximum possible number of interested IMONET members, within a certain time (e.g., the time of the last movie show). Interested users might have an interest in opera, or cinema, or both, and should be located in the "neighborhood" of the theater, so to be able to reach the theater if interested. This type of communication paradigm matches very well with the localized nature of IMONET communications: the information is spread relatively fast in the neighborhood of the sender, while it takes

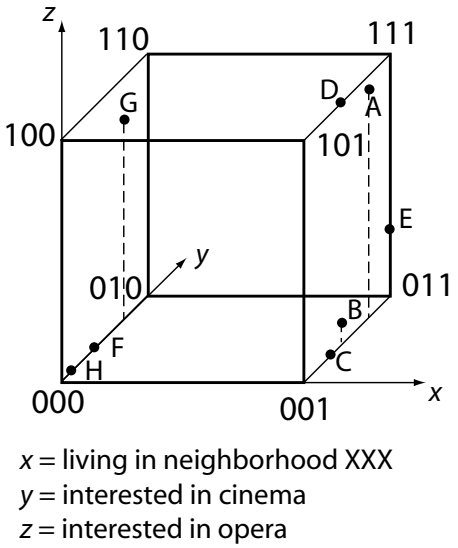


Figure 5.1: An example of network with 3 communities and 8 individuals.

longer time to propagate to remote areas (which are typically less interested in the information, though). Another example of innovative network paradigm for IMONETs might be *community-casting*, in which a certain message is disseminated to all IMONET users who are members of a certain community.

In our work we use the notion of *community profile* to drive information propagation within the IMONET. Assume the network is composed of n individuals, and that the total number of "communities" within the network is m, where m is typically much smaller than n. To simplify presentation, we initially assume that each individual is aware of the total number m of communities within the IMONET, which is a form of global knowledge.

Communities in our approach are intended in a broad sense, and might refer to both an individual's habit (e.g., "individuals living in the same neighborhood", "individuals working in a certain workplace", etc.), and an individual's interests (e.g., "individuals interested in opera", "fan of Bobby Fischer", etc.). An individual community memberships can be determined by either proactively asking the user to indicate her habits and interests, or through extracting information from existing social networking applications (e.g., Facebook), or a combination of the two.

An individual within the network is characterized by his *community profile*, which is

formally defined as follows:

Definition 5.2.1. *The community profile $CP(A)$ of individual A is an m-dimensional vector $CP(A) = (a_1, \ldots, a_m)$, where $a_i \in [0,1]$ is the degree of membership of individual A to the i-th community.*

We decided to associate real, instead of binary, values to individual community memberships to allow better representing different "degrees of interest" in a certain topic (e.g., an individual might be somewhat interested in cinema, and being a passionate opera lover). On the other hand, membership to other types of communities (e.g., "working in a certain workplace") are typically binary.

Given the above definition of community profile, the set of individuals composing an IMONET can be represented as a set of n points in an m-dimensional unit cube, which we call the *community m-cube*. For example, in Figure 5.1 we represents a set of 8 individuals, denoted as A, B, ..., H, in a network with 3 communities. One of the communities – "living in neighborhood XXX" – allows only binary membership values and is represented in the x-axis, whereas the other two – "interested in cinema" and "interested in opera" – have continuous membership values and are represented along the y and z-axis, respectively. In this example, individuals A,...,E live in neighborhood XXX, and have different degrees of interest in cinema and opera (e.g., A is very interested in both, user C is only slightly interested in cinema, user E is very interested in cinema and mildly interested in opera, and so on), while individuals F, G, and H live outside neighborhood XXX.

In the following, we describe a possible realization of two innovative networking services for IMONETs, namely *interest-casting* and *community-casting*. In interest-casting, an IMONET member wants to disseminate a certain information to all (or, the largest possible number of) potentially interested individuals within the network. In community-casting (which can be seen as a particular case of interest-casting), an IMONET member wants to disseminate a certain information to all the members of a specific community.

Let us start with interest-casting, given that community-casting can be seen as a special case of interest-casting. Assume individual A wants to send a message M to all, or the largest possible number of, potentially interested individuals within the network. First, A must define the relative relevance of M to the various communities, which can be done

assigning for each of the m communities a "relevance" value in the $[0,1]$ interval. Such m-dimensional vector associated with a message is called the message *relevance profile*, and is used (coupled with the individuals' community profiles) to drive information propagation within the IMONET. Note that the notion of relevance profile allows to represent message M–similarly to individuals–as a point in the community m-cube. In the following, the relevance profile of message M is denoted $R(M)$. The set of *relevant destinations* for M, denoted $RD(M)$, is the set of individuals within the IMONET for which message M is deemed relevant. The members of this set are the nodes to which M should be delivered, subject to a certain upper bound on the delivery time indicated in the message header called *deadline*. The relevance of M to a certain individual B should be determined considering both the community profile $CP(B)$ of B and the relevance profile $R(M)$ of message M. In our work we use the well-know cosine similarity metric [110] applied to these two vectors, which is defined as follows:

Definition 5.2.2. *Given two m-dimensional vectors A and B, the cosine similarity metric, denote* $\Theta(A,B)$*, is defined as follows:*

$$\Theta(A,B) = \cos(\Theta) = \frac{A \cdot B}{\| A \| \| B \|},$$

where $\| X \|$ *represent the length of vector* X.

Intuitively, the cosine similarity metric expresses the similarity between two vectors in term of the cosine of the angle between them, with value -1 representing exactly opposite, value 0 representing perfect independence, and value 1 representing exactly the same.

Note that, since both individuals' interests and message relevance profiles take values in the unit m-cube, we have that, for any individual B and message M, the angle between $CP(B)$ and $R(M)$ is in $[0,\pi/2]$, implying that $\Theta(B,M)$ is indeed in $[0,1]$. In our work, we use the following simple rule to determine whether message M is relevant to individual B:

Similarity Rule 5.2.1. *The message is relevant if and only if* $\Theta(CP(B),R(M)) \geq \alpha$*, where* α *is a suitably chosen* relevance threshold.

We want to stress the difference between the notion of interest-casting defined herein and more traditional communication paradigms and services such as multi-casting and

publish-subscribe. In interest-casting, the only action taken by a "content provider" (an individual generating a message) is determining the message relevance profile. After that, the message is injected in the network, and information propagation is driven by the notions of relevance and community profile: as we shall see, these notions are used not only to dynamically determine the set of relevant destinations, but also to govern the forwarding process. Thus, in interest-casting the content-provider is not aware of the set of destinations the content should be delivered to, which is in sharp contrast with the traditional notion of multi-casting in which multi-cast groups are explicitly defined and typically known to the content provider. Furthermore, in interest-casting destinations must not explicitly subscribe to a specific "topic", as an individual is able to dynamically "capture" all (or most) relevant messages circulating in the IMONET. This is in sharp contrast with publish-subscribe mechanisms, which typically requires explicit subscription to one or more "topics" to be able to receive relevant information.

In order to reach as many individuals as possible in the set $RD(M)$ or relevant destinations for message M, there is a tradeoff between communication overhead and *coverage*, where by coverage we mean the percentage of the nodes in $RD(M)$ that received M within the time deadline. In other words, given a value for the deadline τ and defined $rD(M,\tau)$ the set of relevant destinations that receive the message M by time τ, we can evaluate the coverage of message M by time τ, which we denote as $\gamma(M,\tau)$, as

$$\gamma(M,\tau) = \frac{|rD(M,\tau)|}{|RD(M)|} \tag{5.2.1}$$

expressed as a percentage value. Note that $rD(M,\tau) \subseteq RD(M)$ for any $\tau \leq \infty$ and that $\lim_{\tau \to +\infty} rD(M,\tau) = RD(M)$.

In order to optimally address this tradeoff, appropriate message forwarding strategies should be defined. A major challenge in defining such strategies is characterizing the relationship between mobility pattern and community profile of an individual, which will be addressed in the remainder of this work.

In case of community-casting, an individual wants to send a message M to all members of a community. In the most general form, membership to a community i can be expressed as "all individuals whose i-th component of the community profile is $\geq \beta$", for

some threshold β. Both community index i and the threshold β used for defining membership are included in the header of the message. The relevant destinations for message M in this case are all individuals whose membership value for community i is at least β. Note that community-casting can be seen as a particular case of interest-casting by setting $R(M) = (0,\ldots,1,0,\ldots,0)$, where the only 1 in $R(M)$ is at position i, and by defining M as relevant for an individual B if and only if $\Theta(R(M),CP(B)) \geq \frac{\beta}{\|B\|}$.

5.3 Community profile and communication opportunities

In this section, we provide a theoretical foundation to our design choice (described in details in the next Section) of using the notion of similarity between relevance and community profiles to drive information propagation within an IMONET.

Let T_{AB} be the random variable representing the time between two consecutive communication opportunities between individuals A and B[1]. Recent studies have demonstrated that T_{AB} (or at least the tail of the distribution) can be modeled quite accurately as an exponential random variable [106]. Let λ_{AB} be the average communication opportunity rate, that is the reciprocal of the average of the duration of the interval between to consecutive communication opportunities between A and B. Our work is based on the assumption, corroborated by sociological studies (see, e.g., [111]), that individuals with common interests meet relatively more frequently than individual with diverse interests. We then assume that λ_{AB} is a function of the community profiles of A and B. In formulas,

$$\lambda_{AB} = \lambda_{AB}\,(CP(A),CP(B)) \tag{5.3.1}$$

To analyze the nature of the relationship between λ_{AB} and the community profiles of A and B, observe that usually our life is organized in activities that derive from our interests or communities. Communication opportunities are the results of such activities. Let us define $T_{AB}^{(i)}$, with $i \leq m$, as the time interval between two consecutive communication opportunities between A and B that occur due to some activity in the scope of community i. Obviously,

[1]Note that, while the focus of this chapter is on communication opportunities arising within the IMONET, our approach can be easily extended to account for more general communication opportunities such as exchanging SMS, social networking software, and so on.

the following relationship holds

$$T_{AB} = \min_{0 \le i \le m} \left\{ T_{AB}^{(i)} \right\} \tag{5.3.2}$$

where $T_{AB}^{(0)}$ represents the time interval between two consecutive communication opportunities that are not due to activities within the scope of the same community. Variable $T_{AB}^{(0)}$ has been introduced to account for occasional communication opportunities (e.g., strangers meeting in a railway station). Note that $T_{AB}^{(0)}$ depends on the mobility system/model in the geographical area of interest.

Assuming that $T_{AB}^{(i)}$ is an exponential random variable as well, and that variables $T_{AB}^{(i)}$ are independent[2], the rate λ_{AB} can be calculated as:

$$\lambda_{AB} = \lambda^{(0)} + \sum_{i=1}^{m} \lambda_{AB}^{(i)} , \tag{5.3.3}$$

where in the above formula we have used the well-known fact that the distribution of the minimum of a set of exponentially distributed, independent random variables with rates $\lambda_1, \ldots, \lambda_k$ is exponentially distributed with rate $\sum_{i=1}^{k} \lambda_i$.

As the frequency of activities within the scope of the i-th community which result in communication opportunities between individuals A and B depend on the degree of membership of A and B to such community, we can write

$$\lambda_{AB}^{(i)} = g_i(a_i, b_i) \tag{5.3.4}$$

where a_i and b_i represent the degree of membership to the i-th community of nodes A and B, respectively, and $g_i(\cdot,\cdot)$ is an appropriate function that increases as a_i and b_i increase. Accurate characterization of the function $g_i(\cdot,\cdot)$ is the subject of future study; nevertheless, intuition suggests that $g_i(\cdot,\cdot)$ should be of type $g_i(a_i, b_i) = a_i\, b_i$ and therefore,

$$\lambda_{AB} = \lambda_0 + k \cdot CP(A) \cdot CP(B) \tag{5.3.5}$$

[2]While we are aware that the $T_{AB}^{(i)}$s might actually be correlated in a practical setting, the one presented in this section should be considered as a first order approximation of the characterization of pairwise communication opportunities rate as a function of their interests.

where k is an appropriate real value that weights the impact of social relationships on the statistics of communications opportunities. In other words, the average rate of communication opportunities between two individuals, say A and B, can be calculated as the sum between a constant λ_0 that characterizes the communication opportunities that do not depend on social relationships between individuals, and a weighted term (the value of the weight is k) that can be evaluated as the scalar product between the community profiles of A and B.

5.4 Community and Interest-based forwarding

In Section 5.2, we have defined two novel communication paradigms for IMONETs, namely interest- and community-casting. In this section, we present a forwarding strategy called CIF (Community- and Interest-based Forwarding) attempting to optimally address the trade-off between communication overhead and coverage. The main idea is to exploit the assumption that individuals sharing interests tend to meet with each other more often than more heterogeneous individuals [111]. Thus, a message M should be preferably forwarded to individuals whose community profile closely resembles the message relevance profile, which is measured by the cosine similarity metric.

In delay tolerant networking individuals can exchange information as a communication opportunity arises. As message delivery is accomplished according to the *store and forward* paradigm, when a communication opportunity occurs between two individuals A and B, the forwarding policy should determine for individual A which messages in her memory buffer should be relayed to individual B, and viceversa. We argue that since social relationships impact the occurrence statistics of the communication opportunities between any two nodes - see eq. (5.3.5), - then the performance of the forwarding policy can be increased by taking the community profiles into account. More specifically, since the communication opportunities become more frequent when individuals share the same interests (their community profiles are similar), then we propose that message M should be relayed to node B if the cosine similarity metric between the relevance values of message M, that we have denoted as $R(M)$, and the community profile of B, that we have denoted as $CP(B)$ is higher

of a given threshold ρ_{Thr} that we call *relaying threshold*, i.e.,

$$\Theta(R(M), CP(B)) \geq \rho_{\text{Thr}} \tag{5.4.1}$$

Note that as the relaying threshold ρ_{Thr} decreases, the forwarding strategy becomes more aggressive: The relaying condition is easily satisfied, thus more and more nodes are choose as relays. This results in the increase of both the coverage of the message M, that we have denotes as $\gamma(M,\tau)$, and the communication overhead (cost) incurred for the delivery of the message M, that we denote as $c(M)$. Observe that the cost $c(M)$ is proportional to the number of copies of the message M spread in the network. Note that the two extreme cases can be considered:

- $\rho_{\text{Thr}} = 0$: in this case all individuals in the network that have a contact with other individuals that store a copy of message M will be appointed to relay the message. In other words, the proposed forwarding strategy becomes the same as Epidemic Forwarding which is the policy reaching the highest coverage value but also the highest cost.

- $\rho_{\text{Thr}} = \alpha$: recall that α is the relevance threshold used to determine the set of destinations of a message; thus, in this case the message is directly delivered by the information source to the relevant destinations as communication opportunities arise between source and destinations. This is the case in which the message delivery cost is minimized along with the message coverage.

In Section 5.5 we will show the impact of the threshold ρ_{Thr} on the performance of the forwarding strategy through numerical examples.

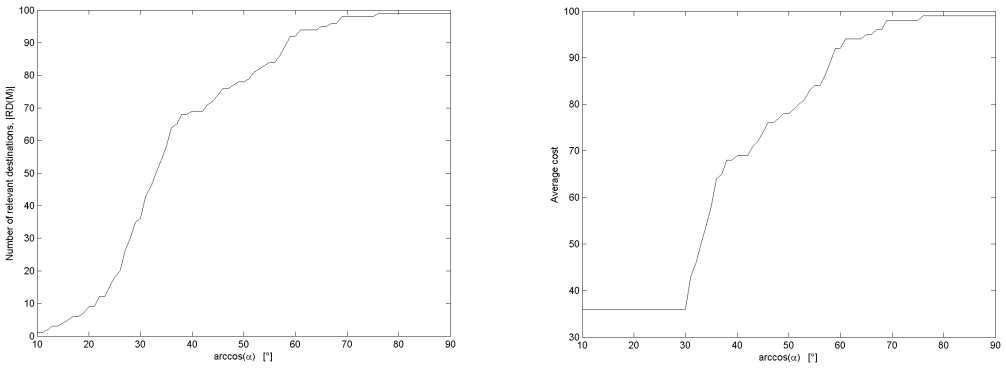
Our proposed forwarding strategy can be justified based on the analysis reported in the previous section as follows. Suppose individual A wants to deliver message M with relevance profile $R(M)$ to the largest possible number of interested individuals. Define a *forwarder* as an individual which is not a destination of M, but that has a copy of M, excluding the source. Our goal is to make the best choice of forwarder nodes. To this purpose, suppose that A can generate only a *limited* number of copies of the message, so that epidemic forward cannot be used. Which is the strategy A should follow in order to

reach the largest possible number of interested individuals, given that only h forwarders can be selected in the network? According to our analysis in the previous section, meeting rates between individuals depend on the scalar product of their profiles–see equation (5.3.5). This readily translates into similarity of their profiles as defined by cosine similarity metric. Thus, individual A might want to forward the h copies of the messages to an individual (say, B) whose community profile $CP(B)$ closely matches $R(M)$, as B is relatively more likely to meet another IMONET member C whose community profile is similar to $CP(B)$. Given transitivity of the similarity metric, individual C in turn is likely to have a profile $CP(C)$ which is close to $R(M)$, thus empirically "maximizing" the probability of reaching an interested individual.

Clearly, the effectiveness of the above described forwarding strategy heavily depends on the actual correlation between pairwise meeting rates and similarity of the respective community profiles, which is modeled by parameter k in equation (5.3.5). In particular, if k becomes very low, occasional meeting opportunities dominate, and a traditional forwarding strategy based on, e.g., selecting as forwarders the first encountered individuals independently of their profile might actually work as well. In the next section, we carefully investigate this issue through simulation.

5.5 Experimental setup and results

In this section we present some experimental results in order to show the performance of CIF. We have implemented both CIF and Epidemic Forwarding in a trace-driven fashion simulator. We have chosen Epidemic Forwarding for performance comparison because to date it is the only existing forwarding solution that can support interest-casting and/or community casting. The traces we use for evaluation have been obtained from the SWIM mobility model, presented in the last chapter. As shown in [1] (and fully discussed in Chapter 4), not only SWIM generates traces that have similar statistical properties to the real ones, but also is a good model in predicting performance of forwarding protocols, including Epidemic. In SWIM, nodes are assigned with a home point on the network area-a square of size $300 \times 300m^2$. Each time a node has to chose its next destination, it trades off distance from its home point and popularity of the place. Thus, nodes that "are



(a) Average value of $RD(M)$ vs. $arccos(\alpha)$ (b) Average cost vs. $arccos(\alpha)$

Figure 5.2: Effect of $arccos(\alpha)$

neighbors" tend to go to the same places, and hence to get in contact more often.

5.5.1 Experimental setup

In order to run CIF on SWIMs traces, we do the following setup: First, we generate a 100 node network. To every node a 4 Dimensional profile vector is given, with entries chosen independently and uniformly at random in $[0,1]$. Each profile vector is then normalized to 1-this way, we make sure that no node has very low interests or no interests at all. We want nodes with similar profile to have a higher meeting rate. Recall that in SWIM, "neighbors" (nodes whose home point is not far) tend to have more communication opportunities. Thus, a high correlation between home points and node profiles yields a high meeting rate between nodes with similar profile. So, for every node we derive its home point on the 2 Dimensional network area through a linear mapping, in such a way that the correlation between the cosine similarity of two profile and the distance of the two corresponding home points is high (in our case it is -0.896732). Then we generate mobility traces using SWIM as the mobility model.

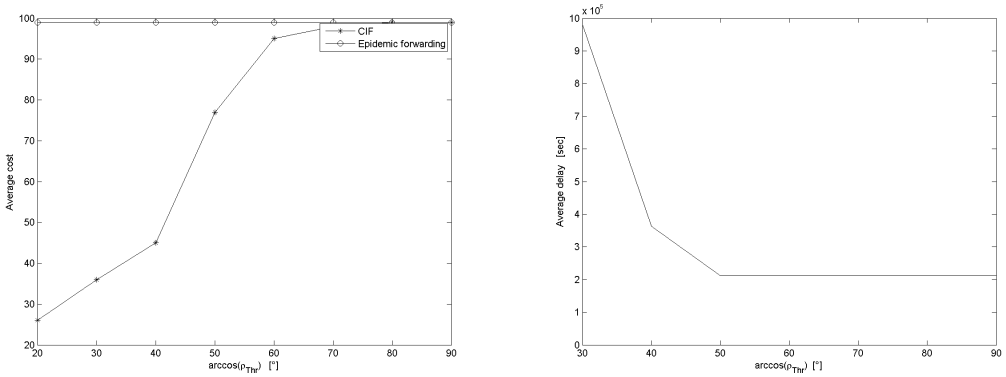
5.5.2 Results

Here we present performance results of CIF and compare it with Epidemic Forwarding. We pick randomly a node (individual) in the network, and let it be the source of the message to be forwarded. Then we generate a message of the same profile of the source node, and wait till it is forwarded to as many relevant destinations as possible, using both Epidemic forwarding and CIF. The metrics we are concerned with are: *cost*, which is the number of replicas per generated message; *coverage*, which is the number of relevant destinations reached by the message, and *average delay*, which is the average duration from the moment a message is generated till it reaches as many relevant destinations as possible. The relevant destinations are determined from the relevance threshold α as explained in Section 5.2. This is confirmed in Figure 5.2(a) where we show the average[3] number of relevant destinations versus the arc-cosine of the relevance threshold. As expected, as α decreases, and therefore the angle $arccos(\alpha)$ between community profiles of nodes and the relevance profile of the message increases, the number of relevant destinations increases. Accordingly, also the cost depends on the value α. In fact, the lower α–which means the higher $arccos(\alpha)$, the higher the number of copies of the message which will be generated. This is confirmed in Figure 5.2(b) where we show the average cost versus the value of the arc-cosine of the relevance threshold α when the relaying threshold is $\rho_{Thr} = cos(30)$.

Once the value of the relevance threshold α has been set, the performance depends on the value of the relaying threshold ρ_{Thr}. In Figure 5.3(a) we show the average cost when the relevance threshold is $\alpha = 0.94$, versus the value of the relaying threshold ρ_{Thr}. As expected the communication cost increases as $arccos(\rho_{Thr})$ increases. This is straightforward as an increase in this parameter results in a decrease in ρ_{Thr} and therefore in a less selective relaying policy. In the same curve we represent the cost for Epidemic Forwarding which is always equal to 99, given that a copy of the message will be generated for each node in the network.

It is evident that CIF achieves much lower cost than epidemic forwarding, like it was expected. Such performance gain is paid off in terms of delay. In Figure 5.3(b) we show the

[3]The values shown in this section have been obtained as the average on 10 simulations



(a) Average cost vs. $arccos(\rho_{Thr})$ (b) Average delay vs. $arccos(\rho_{Thr})$

Figure 5.3: Effect of $arccos(\rho_{Thr})$

average delay needed to deliver the message to all relevant destinations dependent on the relaying threshold. We observe that the delay is very high when the value of $arccos(\rho_{Thr})$ is low, and therefore the relaying policy is very restrictive which results in low cost. On the contrary, high values of $arccos(\rho_{Thr})$ corresponding to low relaying threshold ρ_{Thr}, make the relaying policy more aggressive. Thus, the delay decreases significantly.

It might be interesting to observe how the message coverage increases with the time. Accordingly, in Figure 5.4 we show the value τ when a certain coverage $\gamma(M,\tau)$ of the message is reached. As expected the time required to reach a certain coverage value decreases as the value of $arccos(\rho_{Thr})$ increases. From the above discussions it is evident that protocol designers are required to select the most appropriate tradeoff between cost and delay and set the relaying threshold, ρ_{Thr} accordingly.

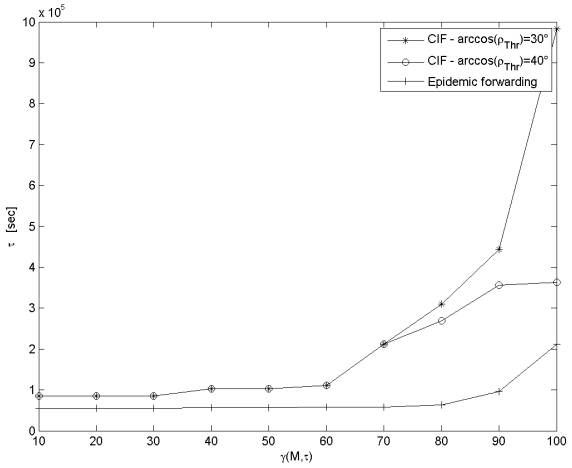


Figure 5.4: Value of τ versus the message coverage $\gamma(M,\tau)$

.

5.6 Conclusions

Despite existing work in the area of PSNs, we believe that substantial efforts are still needed to really understand the potential of these networks, in particular in terms of the innovative services, applications, and communication paradigms that could be realized therein. Here we aimed at making a few steps forward in uncovering this potential, by introducing the IMONET paradigm. We then observed that, while the analysis of social-ties between the individuals forming an IMONET has been extensively used to optimize performance of *traditional* communication paradigms and services (e.g., end-to-end routing, multicasting, and publish-subscribe), the characterization of an individual as a member of a social structure has not yet been used to design *innovative* communication paradigms and services *specifically designed* for an IMONET user needs.

We advocated this change of perspective, and proposed two innovative communication paradigms for IMONETs, namely *interest-casting* and *community-casting*. In interest-casting, a message generated by an IMONET member is selectively forwarded in the network with the goal of reaching potentially interested members. In community-casting, a message generated by an IMONET member is selectively forwarded to the IMONET nodes who are members of a certain community (typically, one of the communities to which the sender node belongs). To the best of our knowledge, these are the first communication

paradigms specifically designed taking into account IMONET user needs.

To address the scalability issue, we started observing that, although distributed community detection is in principle possible [78], dynamically detecting the structural features of social ties between members is likely to become impractical when the size of the network grows beyond a few hundreds nodes. Thus, approaches based on tracking pair-wise contact patterns between members to deduce social ties (like, e.g., [108, 11, 106, 12, 105]) are doomed to perform poorly in medium- to large-scale IMONETs. To circumvent this problem, we let an IMONET member explicitly characterizes him/herself as an individual with a certain set of habits and interests through the novel concept of *community profile*. The basic idea exploited in both interest- and community-casting is that when two individuals have a communication opportunity, they exchange each other community profile, as well as the *relevance profile* of the messages they are carrying in the respective buffers. Based on this limited information exchange, the nodes decide which messages (if any) to move/copy into each other buffer. Note that this basics mechanism is *stateless*, hence appropriate for implementation even in large-scale IMONETs.

After the definition of interest- and community-casting, we presented the complete design of CIF, a community and interest-based forwarding mechanism based on the well-known cosine similarity metric [110]. We showed through extensive simulation based on a realistic individual mobility model that the proposed CIF protocol is able to disseminate information to interested IMONET members almost as quickly as epidemic dissemination, but with a much reduced overhead.

We believe these paradigms, possibly complemented with other paradigms yet to be defined, can be considered as a starting point for the development of innovative IMONET services/applications, thus turning the IMONET vision into reality.

Future work and concluding remarks

The recent achievements of technology in yielding affordable, small, powerful, multitasking devices such as the last generation of lap tops, cellphones, PDAs, etc., not only has made our life easier but also has raised the need for new innovative and user-oriented applications. At the same time, wireless communication between these devices weakens the underlying network by making it vulnerable from all sort of attacks, including jamming. In this dissertation we started with discussing the routing techniques in this context, and presented an elegant and efficient way able to overcome the congestion problem in large ad-hoc wireless networks. Our *routing in outer space*, used as a top-layer of every position-based routing protocol yields a network that is more secure and, at the same time, more energy-efficient. In our study we have considered a squared networked area, but routing in outer space can be adapted to other bi-dimensional surfaces by approximating them with the smallest outer square. Clearly the approximation will not yield route distribution as perfect as in the case of the square. As future work we would like to apply techniques similar to the ones of routing in outer space to other bi-dimensional areas and to other symmetric spaces (such as sphere for example), and see if we can get good results in route distribution and maybe better ones in terms of route stretch, thus in energy efficiency.

Routing in outer space is the first proactive protocol proven to guarantee global load balancing among nodes in the network. It is based on greedy forwarding of messages, hence it doesn't guarantee local load balancing in single nodes' neighborhoods. As we already mentioned there are several approaches that try to solve the congestion issue locally, like [35, 48, 49], in a reactive way. Clearly these approaches can be used with routing in outer space in a higher layer, in order to smooth the problem as soon as it arises. Another direction of study in this area that we would like to explore is adapting the virtual mapping

of the "outer space" technique to solve this problem in a proactive way.

The second problem considered in this dissertation is related to the forwarding techniques in social mobile wireless networks, where we discussed the need for new protocols that exploit "good" characteristics of human behavior in strengthening the network from "bad" sides of human personality, such as *selfishness*. We presented the first forwarding protocols for social mobile wireless networks, *Give to Get Epidemic* and *Give to Get Delegation*, that exploit social aspects of the network in order to detect misbehaving individuals, and in the same time, are more efficient in terms of message replicas then their vanilla alter egos: Epidemic and Delegation.

We continued our study of the social mobile wireless networks and presented SWIM, the first mobility model that besides from being simple, generates synthetic traces similar to the real ones and shows accuracy in predicting the performance of forwarding protocols. The model is based on simple observations from the everyday life: We tend to go to nearby places that are popular. SWIM is the first model to show experimentally and theoretically the presence of the power-law and exponential dichotomy of inter-contact time distribution. It generates simulated scenarios that match the real ones in terms of inter-contact time, contact duration and number of contacts between nodes. As future work in this direction we would like to study how the prevalence of site popularity/distance from home-point in nodes' decisions affects the system. We are interested in uncovering possible complex sub-structures in the network contact graph such as communities. Also we want to see how the initial distribution of home-points on the network area influences these sub-structures and network properties in general. Finally we would like to study other statistical metrics of the network such as inter-arrival times and node distribution in sites, moving distances distribution etc., and to see if any of them follows a stationary distribution.

The goodness of SWIM turns it in a useful tool in protocol validation for social mobile networks and provides further insights towards the understanding of human mobility: The individual, his needs and habits should be put in the center of the study. With this in mind we introduced the concept of INdividual MObile NETworks (IMONETS), where each individual is associated with a *community profile* that characterizes him with a certain set of habits, interests and social relations within the network. We then presented a community

and interest casting protocol called CIF, that is able to disseminate information throughout the network targeted to two different type of individuals: The ones that belong to a certain community/group (e.g. people that work at Sapienza University of Rome), or people that share certain interests (e.g. chess players). With the use of SWIM we show that CIF is able to reach the wanted "audience" almost as quickly as Epidemic Forwarding, but with a much lower communication overhead.

Though ad-hoc wireless networks have been fully studied in the last years, there are still lots of open problems in this area. The aim of this dissertation was to attack and solve some of the central problems in this context, by using original and elegant ideas, such as in routing in outer space, that could be further investigated to create innovative techniques. At the same time, we wanted to raise new questions and give insights towards better understanding of human mobility and of social mobile wireless networks, as it seems to be a very promising technology that can support revolutionary services. We hope to have reached our goal.

Bibliography

[1] A. Mei and J. Stefa. Routing in Outer Space. In *INFOCOM 2008: Proceedings of The 27th IEEE Conference on Computer Communications (mini-conference)*, pages 2234–2242, April 2008.

[2] A. Mei and J. Stefa. Routing in Outer Space: Fair Traffic Load in Multi-Hop Wireless Networks. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 23–32, New York, NY, USA, 2008. ACM.

[3] A. Mei and J. Stefa. Routing in Outer Space: Fair Traffic Load in Multi-Hop Wireless Networks. *IEEE Transactions on Computers*, 58(6):839–850, 2009.

[4] A. Mei and J. Stefa. SWIM: A Simple Model to Generate Small Mobile Worlds. In *INFOCOM '09: Proceedings of The 28th IEEE Conference on Computer Communications*, pages 2106–2113, April 2009.

[5] A. Mei and J. Stefa. Give2Get: Forwarding in Social Mobile Wireless Networks of Selfish Individuals. *Submitted for publication to international conference*, 2009.

[6] A. Mei, G. Morabito, P. Santi, and J. Stefa. Stateless Interest- and Community-based Information Dissemination in Individual MObile NETworks. *Submitted for publication to international conference*, 2009.

[7] S. Kwon and N. B. Shroff. Paradox of shortest path routing for large multi-hop wireless networks. In *INFOCOM '07: Proceedings of the 26th IEEE International Conference on Computer Communications*, May 2007.

[8] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM Press, 2005.

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

[10] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008.

[11] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007.

[12] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008.

[13] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[14] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *INFOCOM '06. Proceedings of the 25th IEEE International Conference on Computer Communications*, 2006.

[15] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 183–194. ACM, 2007.

[16] H. Cai and D. Y. Eun. Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 159–170. ACM, 2007.

[17] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communcations Magazine*, 40:102–114, August 2002.

[18] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM Press.

[19] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94: Proceedings of the Conference on Communications Architectures, Protocols, and Applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.

[20] S. Murthy and J. J-Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[21] W. Liu, C. Chiang, H. Wu, and C. Gerla. Routing in clustered multihop mobile wireless networks with fading channel. In *Proceedings of IEEE SICON'97*, pages 197–211, April 1997.

[22] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[23] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 1405, Washington, DC, USA, 1997. IEEE Computer Society.

[24] C. E. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[25] Z. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of the IEEE Int. Conf. on Universal Personal Communications*, October 1997.

[26] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, 1998.

[27] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks, April 1999.

[28] Rahul Jain Anuj, Rahul Jain, Anuj Puri, and Raja Sengupta. Geographical routing using partial information for wireless ad. *IEEE Personal Communications*, 8:48–57, 1999.

[29] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, October 1997.

[30] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom 2000: Proceedings of the 6th annual ACMinternational conference on Mobile computing and networking*, 2000.

[31] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. Self–configuring localization systems: Design and experimental evaluation. *Trans. on Embedded Computing Sys.*, 3(1):24–60, 2004.

[32] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grain localization in ad-hoc networks of sensors. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 2001.

[33] S. Capkun, M. Hamdi, and J-P. Hubaux. GPS-free Positioning in Mobile Ad Hoc Networks. *Cluster Computing*, 5(2):157–167, 2002.

[34] L. Lazos and R. Poovendran. Serloc: secure range-independent localization for wireless sensor networks. *ACM Transactions on Sensor Networks*, 2:325–358, August 2006.

[35] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department, May 2001.

[36] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01)*, Rome, Italy, July 2001.

[37] S. Basagni, M. Nati, and C. Petrioli. Localization error-resilient geographic routing for wireless sensor networks. In *Proceedings of IEEE Global Communications Conference (GLOBECOM'08)*, December 2008.

[38] L. Blazevic, L. Buttyán, S. Capkun, S. Giordano, J-P. Hubaux, and J-Y. LeBoudec. Towards self-organized mobile ad hoc networks: The terminodes project. *IEEE Communications Magazine*, June 2001.

[39] L. Blazevic, S. Giordano, and J-Y. LeBoudec. Self organized terminode routing. *Cluster Computing Journal*, April 2002.

[40] K. Seada and A. Helmy. Geographic protocols in sensor networks. Technical report, USC, July 2004.

[41] P. P. Pham and S. Perreau. Increasing the network performance using multi-path routing mechanism with load balance. *Ad Hoc Networks*, 2:433–459, October 2004.

[42] Y. Ganjali and A. Keshavarzian. Load balancing in ad hoc networks: single-path routing vs. multi-path routing. In *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, volume 2, pages 1120–1125 vol.2, 2004.

[43] B. Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference*

on Mobile Computing and Networking (MobiCom '00), Boston, MA, USA, August 2000.

[44] V. Srinivasan, P. Neggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc wireless networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 2003.

[45] W. Wang, X. Li, and Y. Wang. Truthful multicast routing in selfish wireless networks. In *Proceedings of the 10th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '04)*, Philadelphia, PA, USA, September 2004.

[46] E. Hyytiä and J. Virtamo. On traffic load distribution and load balancing in dense wireless multihop networks. In *NGI 2006*, 2006.

[47] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica. Balancing traffic load in wireless networks with curveball routing. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 170–179, New York, NY, USA, 2007. ACM.

[48] M. Zorzi and R. R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: Energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.

[49] P. Casari, M. Nati, C. Petrioli, and M. Zorzi. Efficient non-planar routing around dead ends in sparse topologies using random forwarding. In *Proceedings of IEEE International Conference on Communications (ICC '07)*, pages 3122–3129, June 2007.

[50] Q. Fang, J. Gao, and L. J. Guibas. Locating and bypassing holes in sensor networks. *Mob. Netw. Appl.*, 11(2):187–200, 2006.

[51] A. Kröller, S. P. Fekete, D. Pfisterer, and St. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1000–1009, New York, NY, USA, 2006. ACM Press.

[52] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 122–133, New York, NY, USA, 2006. ACM Press.

[53] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), March 2000.

[54] A. Zemlianov and G. de Veciana. Capacity of ad hoc wireless networks with infrastructure support. *IEEE Journal on Selected Areas in Communications*, 23(3), March 2005.

[55] X. Hong, P. Wang, J. Kong, Q. Zheng, and J. Liu. Effective probabilistic approach protecting sensor traffic. *Military Communications Conference, 2005. MILCOM 2005. IEEE*, 1:169–175, October 2005.

[56] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office, 1997.

[57] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, August 2000.

[58] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[59] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108, New York, NY, USA, 2003. ACM.

[60] M. Bhardwaj and A. P. Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, volume 3, pages 1587–1596, 2002.

[61] D. M. Blough and P. Santi. Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in ad-hoc networks. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (ACM MobiCom 2002)*, 2002.

[62] H. Zhang and J. Hou. On deriving the upper bound of α—lifetime for large sensor networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 121–132, New York, NY, USA, 2004. ACM Press.

[63] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *Mobile Computing, IEEE Transactions on*, 2(3), July-Sept. 2003.

[64] J. Le Boudec and I. Vojnovic. Perfect simulation and stationery of a class of mobility models. In *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM'05)*, March 2005.

[65] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. of the IEEE Security and Privacy Symposim 2003.*, 2003.

[66] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis and defenses. *Proc. of 3rd IEEE/ACM Information Processing in Sensor Networks (IPSN'04)*, pages 259–268, 26-27 April 2004.

[67] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 49–63, Washington, DC, USA, 2005. IEEE Computer Society.

[68] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei. A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In *Proceedings of the Eighth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2007)*, pages 80–89, 2007.

[69] K. Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34, 2003.

[70] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 237–243. ACM, 2005.

[71] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[72] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/cambridge (v. 2006–01–31). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/cambridge, January 2006.

[73] J. Leguay, A. Lindgren, J. Scott, T. Riedman, J. Crowcroft, and P. Hui. CRAWDAD trace upmc/content/imote/cambridge (v. 2006–11–17). Downloaded from http://crawdad.cs.dartmouth.edu/upmc/content/imote/cambridge, nov 2006.

[74] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006–01–31). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom, January 2006.

[75] 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.

[76] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 183–194. ACM Press, 2007.

[77] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–782, june 2008.

[78] P. Hui, E. Yoneki, S.Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *MobiArch '07: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, pages 1–8, New York, NY, USA, 2007. ACM.

[79] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043), 2005.

[80] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM.

[81] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. Bar fault tolerance for cooperative services. In *In SOSP05 20th ACM Symposium on Operating Systems Principles*. ACM, 2005.

[82] H. C. Li, , A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. Bar gossip. In *Proceedings of the 7th Symposium on Operating System Design and Implementation (OSDI '06)*, 2006.

[83] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, New York, NY, USA, 2000. ACM.

[84] D.B. Johnson. Routing in ad hoc networks of mobile hosts. In *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pages 158–163, Dec 1994.

[85] S. Buchegger and J-Y. Le Boudec. Performance analysis of the CONFIDANT protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks. In *MobiHoc*

02: Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, June 2002.

[86] K. Balakrishnan, Jing Deng, and V.K. Varshney. Twoack: preventing selfishness in mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, March 2005.

[87] L. Buttyán and J-P. Hubaux. Enforcing service availability in mobile ad-hoc wans. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. IEEE Press, 2000.

[88] H. Miranda and L. Rodrigues. Preventing selshness in open mobile ad hoc networks. In *Proceedings of the Seventh CaberNet Radicals Workshop*, October 2002.

[89] L. Buttyán and J-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mob. Netw. Appl.*, 8(5):579–592, 2003.

[90] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *INFOCOM 2003. IEEE International Conference on Computer Communications. Proceedings*, 2003.

[91] Levente Buttyán, László Dóra, Márk Félegyházi, and István Vajda. Barter trade improves message delivery in opportunistic networks. *Ad Hoc Networks*, 8(1):1–14, 2010.

[92] SUMO-Simulation of Urban MObility. http://sumo.sourceforge.net/.

[93] P. Hui, K. Xu, V. Li, J. Crowcroft, V. Latora, and P. Lio. Selshness, altruism and message spreading in mobile social networks. In *Proceeding of First IEEE International Workshop on Network Science For Communication Networks (NetSciCom09)*, April 2009.

[94] A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track*, 2008.

[95] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[96] CRAWDAD: Community Resource for Archiving Wireless Data At Dartmouth. http://crawdad.cs.dartmouth.edu/.

[97] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *INFOCOM 2008. IEEE International Conference on Computer Communications. Proceedings*, 2008.

[98] M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(3):59–70, 2007.

[99] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. On Clustering Phenomenon in Mobile Partitioned Networks. In *The First ACM SIGMOBILE International Workshop on Mobility Models for Networking Research*. ACM, 2008.

[100] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *The First ACM SIGMOBILE International Workshop on Mobility Models for Networking Research*. ACM, 2008.

[101] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft. Opportunistic content distribution in an urban setting. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*. ACM, 2006.

[102] H. Cai and D. Y. Eun. Toward stochastic anatomy of inter-meeting time distribution under general mobility models. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 273–282. ACM, 2008.

[103] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *MobiArch '07: Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture*, pages 1–8. ACM, 2007.

[104] E. Yoneki, P. Hui, S. Y. Chan, and J. Crowcroft. A socio-aware overlay for publish/-subscribe communication in delay tolerant networks. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 225–234. ACM, 2007.

[105] F. Li and J. Wu. Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pages 1–9, June 2009.

[106] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 299–308, New York, NY, USA, 2009. ACM.

[107] S. Ioannidis, A. Chaintreau, and L. Massoulié. Optimal and scalable distribution of content updates over a mobile social network. In *Proceedings of INFOCOM 2009*, pages 1422–1430, April 2009.

[108] C. Boldrini, M. Conti, and A. Passarella. Contentplace: social-aware data dissemination in opportunistic networks. In *MSWiM '08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 203–210, New York, NY, USA, 2008. ACM.

[109] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):748–760, June 2008.

[110] M.M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, Berlin, 2009.

[111] M. Mcpherson, L.S. Lovin, and J.M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.