Exploiting Social Interactions in Mobile Systems

Andrew G. Miklas[1], Kiran K. Gollu[1], Kelvin K.W. Chan[2], Stefan Saroiu[1], Krishna P. Gummadi[3], and Eyal de Lara[1]

[1] University of Toronto
[2] Google
[3] MPI for Software Systems

Abstract. The popularity of handheld devices has created a flurry of research activity into new protocols and applications that can handle and exploit the defining characteristic of this new environment – user mobility. In addition to mobility, another defining characteristic of mobile systems is user social interaction. This paper investigates how mobile systems could exploit people's social interactions to improve these systems' performance and query hit rate. For this, we build a trace-driven simulator that enables us to re-create the behavior of mobile systems in a social environment. We use our simulator to study three diverse mobile systems: DTN routing protocols, firewalls preventing a worm infection, and a mobile P2P file-sharing system. In each of these three cases, we find that mobile systems can benefit substantially from exploiting social information.

1 Introduction

Recent news articles are reporting a dramatic increase in the use of battery-powered, mobile, lightweight, handheld devices often equipped with wireless interfaces [13,4]. Examples of such ubiquitous devices include cell-phones and PDAs, music players like Zune, and gaming devices like PSP. The number of mobile systems for these devices is also quickly growing. Their key challenge is providing functionality in a dynamic and often unreliable network environment. This need has led to a flurry of research on the design and implementation of new protocols and applications that can handle (and perhaps exploit) the primary characteristic of this new environment – user mobility.

In addition to user mobility, another defining characteristic of mobile systems is user social interaction. A variety of new applications focus on facilitating social activities in pervasive systems. For example, new Internet dating services allow clients to use their cell-phones' Bluetooth radios to detect when they are in the proximity of a person that matches their interests [18]. Other companies are offering file-sharing software for mobile phones that allows users to share ring-tones, music, games, photos, and video [28,17]. In these new mobile systems, information exchange is driven by the users' social interactions: friends use their cell-phones to share photos or song collections; strangers with similar dating profiles are notified when they are near each other.

In this paper, we examine how these mobile systems could exploit people's social relations to make more informed decisions, potentially leading to substantial performance gains and higher query hit rates. We start by classifying social interactions in two categories. One category is interactions between friends, that is people who meet

J. Krumm et al. (Eds.): UbiComp 2007, LNCS 4717, pp. 409–428, 2007.

more regularly and for longer periods of time. The other category is interactions between strangers, that is people who meet sporadically, by passing each other by. Note that in practice, the spectrum of social interactions is quite complex. For instance, a pair of people could be classified as "familiar strangers" [22] – two people encountering regularly without ever interacting or forming an explicit relationship of a social nature. Nevertheless, in this paper, we classify all relationships only as friends or as strangers; based on our simple definitions, we classify familiar strangers as friends. We leave a more complex social classification to future work.

We investigate the potential of incorporating social information in three mobile systems with diverse characteristics. First, we study the performance of routing protocols in delay tolerant networks (DTNs) when a sender and a receiver are friends, and when they are strangers. Our findings show that incorporating social information in routing decisions significantly improves the performance of several DTN routing protocols. Second, we examine whether firewalls that discriminate between traffic sent by friends and traffic sent by strangers can slow down the propagation of a worm or virus in a mobile network. We find that worms spread significantly slower if a small fraction of nodes reject traffic sent by strangers. Third, we examine the performance of file exchange protocols in a P2P file-sharing application. We find that sharing files only among friends drastically reduces the rate of successful requests in such systems. To maintain a high query hit rate, mobile P2P systems must allow their users to exchange content with strangers. In summary, we show that separating people's interactions only as friends and strangers leads to a more efficient routing protocol, a more effective security measure, and a higher query hit rate in a mobile application.

We build a trace-driven simulator that enables us to re-create the behavior of mobile systems in a social environment. Our simulator recreates all encounters between a large population of mobile users. To build our simulator, we analyze a 101-day trace of encounters between people equipped with Bluetooth-enabled cell-phones collected by the "Reality Mining" project at the MIT Media Lab [23]. To generate encounters between friends, we use a well-known social networking model – the Watts-Strogatz model [33]. To generate encounters between strangers, our simulator uses a heavy-tailed model inspired from the well-known preferential attachment model [3]. By combining encounters between friends and encounters between strangers, we can accurately simulate how social information can lead to performance gains and higher query hit rates in our three mobile systems.

The paper is organized as follows. Section 2 presents our trace-based analysis of people encounters. Section 3 uses our observations and analysis to develop a social networking-based simulator of people encounters. In Section 4, we use our simulator to study the effect of incorporating social information to three mobile systems: DTN routing, the spread of worms in a mobile network, and the performance of file-sharing applications. Section 5 summarizes our results and presents conclusions.

2 Characterizing People's Encounters

To perform an evaluation of using social information in mobile systems, we need a data trace of a mobile environment together with information about the social relationship among the participants. Unfortunately, we are unaware of any such previously gathered

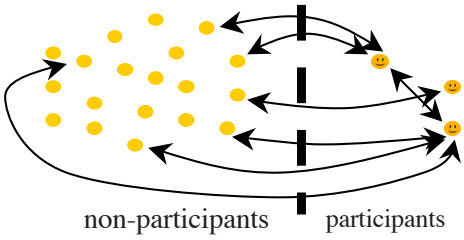

Fig. 1. The type of encounters present in the trace. One arrow represents one encounter. Each pair of people could have more than one encounter. Encounters between non-participants are not captured in the trace.

traces. Instead, we perform a social-based analysis of a trace of Bluetooth activity to annotate it with the required information. For this, we use a 101-day trace of encounters between people equipped with Bluetooth-enabled cell-phones collected by the "Reality Mining" project at the MIT Media Lab [23]. By studying the frequency of encounters, we can annotate this trace with social information by classifying pairs of people who encounter frequently as "friends", whereas pairs of people encountering sporadically are classified as "strangers". The Reality Mining group has also used this trace to infer social relationships between participants. Their analysis is focused on identifying different contexts in which social relationships are formed. Instead, our goal is to characterize the key temporal and social parameters of people's encounters from this trace.

2.1 Trace Description

Gathering a suitable trace to analyze the properties of people encounters is very challenging. Such a trace requires tracking many people simultaneously while recording all interactions among them. Collecting the data must not inconvenience the individuals being monitored and tracked. The privacy concerns raised by such experiments makes it particularly difficult to gather the data at scale. For all these reasons, very few large-scale traces of people encounters are available.

We use a trace collected by the Reality Mining project at the MIT Media Lab [23]. This project equipped 100 students with Bluetooth-enabled cell-phones. The phones were instrumented to probe and discover all nearby Bluetooth devices every five minutes. Data was collected for the entire 2004 – 2005 academic year producing a trace with over 285,000 Bluetooth-to-Bluetooth contacts.

We use this data as a rough approximation of people encounters since most of the Bluetooth-to-Bluetooth contacts involve people encounters. Many participants used the instrumented devices as their primary cell-phones. Consequently, these cell-phones were able to capture these individuals' encounters across a broad range of their day-to-day activities; the trace is not limited to the time that participants spent on campus or in their lab only.

While the trace captures all encounters between participants themselves, the majority of encounters present in the trace are between participants and non-participants. A non-participant appears in the trace whenever their cell-phone responded to Bluetooth

Table 1. Summary statistics for trace of people encounters, 09/08/2004 to 12/17/2004. Each participant encounters other people, either participants or non-participants. One pair of people can encounter each other multiple times.

data source	Bluetooth cell-phones
trace length	101 days, 0 hours, 49 mins
participants	88
non-participants	10,739
total # of encounters	155.321
# of pairs of people encountering	28,166
median # encounters per participant	1,970

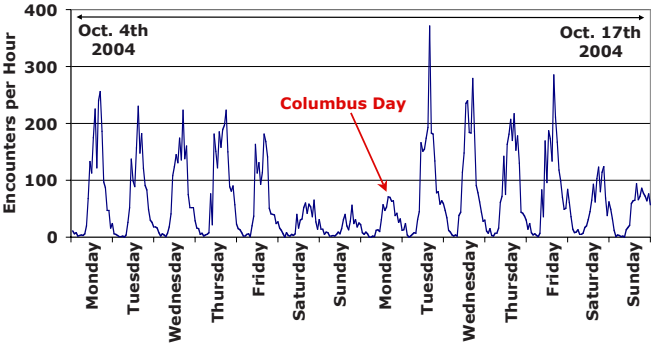

Fig. 2. The number of encounters over a two week period. Encounters show diurnal and weekly patterns. This two week period includes a U.S. statutory holiday.

probes from a participant's instrumented phone. This data gives us only a partial view into the behavior of non-participants: we lack additional information on how they encounter each other. While all encounters with non-participants are included in the study, our analysis's findings are restricted to the set of participants only. Figure 1 illustrates the type of encounters present in the trace.

The use of only one trace in our analysis restricts the applicability of our conclusions to the general population. This problem is further exacerbated by the limited scope of the sample population; it consists entirely of students, professors, and other academic staff. We hope to validate our findings with larger scale traces conducted in a variety of contexts as they become available.

2.2 High-Level Trace Statistics

In all our analysis, we use a trace of people encounters that spans the Fall school term only. Table 1 shows the summary statistics of the trace we used. The trace contains over 155K encounters made by 88 participants over 101 days. On average, there is one encounter every 7 seconds. The peak rate of encounters in the trace is 370 encounters over 10 minutes, while the longest period with no encounters reported is 4 hours and 24 minutes.

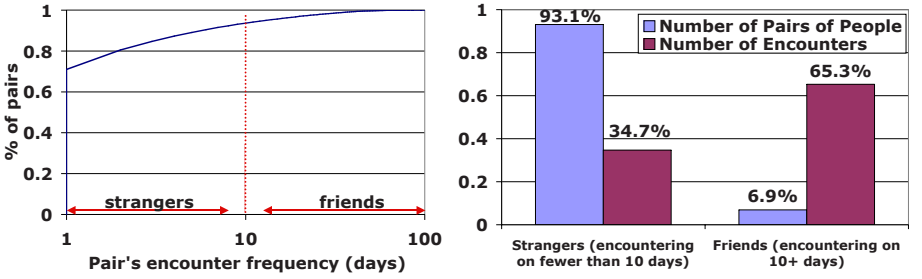

Fig. 3. There are two types of pairs of people: friends and strangers. (a) CDFs of number of pairs of people as a function of the pairs' encounter frequency, and (b) number of pairs and number of encounters as a function of the pairs' encounter frequency, split in two groups.

Figure 2 shows the number of encounters per hour for a typical two week period. As expected, encounters show diurnal and weekly patterns. The two week period shown includes a statutory U.S. holiday (Columbus Day) that shows the same level of activity as a typical day on a week-end. We checked the MIT school calendar; the school is officially closed during Columbus Day.

2.3 Two Types of People Encounters: Friends and Strangers

We would like to investigate how people's social relations affect their encounters. For this, we use the number of days on which two people encounter as a first-degree approximation of their social relation. Intuitively, people encountering on many different days are likely to have a strong social relation (i.e., they are friends) as opposed to people who rarely encounter (i.e., they are strangers).

Figure 3a shows the percentage of pairs of people with encounters as a function of their encounters frequency. The graph shows that most pairs of people (71%) encounter on only one day. Less than 7% of pairs encounter on 10 or more days. We classify encounters into two groups: between pairs of people who encountered on fewer than 10 days in our trace, and between pairs of people who encountered on at least 10 days. We chose the value 10 days as a reasonable lower bound for the number of days on which two friends encounter in the trace if they were to meet weekly. Our trace spans 14 full weeks.

Figure 3b shows the number of pairs and the number of encounters broken down by their types: friend versus stranger encounters. While only 6.9% of pairs of people were friends, these pairs account for two-thirds (65.3%) of all encounters in the trace. This demonstrates that while most pairs of people encountering are strangers having no social relation, most encounters made are between friends. Thus, if our concern is to propagate information quickly across a mobile network, we need to focus on stranger encounters since they are rare opportunities for different people to exchange information. However, if our concern is to provide more stable and predictable network links for an application, then we must focus on friend encounters.

The stark difference between friend encounters and stranger encounters lead us to study their properties independently for much of the analysis that follows.

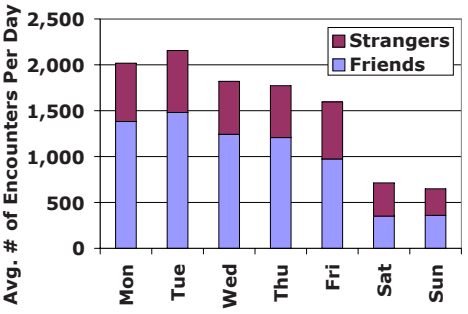

Fig. 4. Daily encounters. The average number of encounters per day broken down by day-of-the-week. People have more encounters during week days than week-end days. Two thirds of the daily encounters are with friends.

2.4 Weekly and Diurnal Patterns

As previously shown in Figure 2, people encounters present weekly and daily patterns. In this subsection, we take a closer look at the day-of-the-week and time-of-the-day effects present in the trace.

Figure 4 shows the average number of daily encounters broken down by the day-of-the-week when they occur. While more encounters occur on week days than on week-end days, the number of encounters is roughly the same across all week days. This suggests that people's behavior is consistent across each day of the week and across each day of the week-end. Figure 4 also separates friend encounters from stranger encounters. For each day of the week, two thirds of encounters (between 61 and 68%) are friend encounters and one third are stranger encounters. Over the week-end, this behavior is more balanced, only 50 to 55% of encounters are friend encounters.

We also examine the number of daily encounters by hour-of-the-day for both week days and week-end days (these results are not graphed for lack of space.) We find that most people's encounters occur on afternoons during week days with a peak at 4:00pm. There are 50% more encounters on afternoons (2-5pm) compared to mornings (9am-12pm). The diurnal pattern of week-end days is different than that of week days: week-ends have high activity during late afternoons and even late nights, but relatively little activity during mornings.

To understand whether people's encounter rates are predictable, we first calculated each participant's rate of encounters for each hour of the day. For each individual, we measured how consistent their encounter rate is during the same hour across all week days and across all week-end days. For example, we measure how often the number of encounters between 1pm and 2pm on Monday through Fridays change. We consider Saturday and Sunday separately since week-ends have a different dynamic of how people encounter. For each pair of consecutive hour slots, we compute the difference in the number of events for each individual.

Figure 5 shows the distribution of the differences of an individual's number of encounters for the same hour-of-the-day for week days and week-end days. From this graph, we can see that people's encounter rates are predictable. On average, an

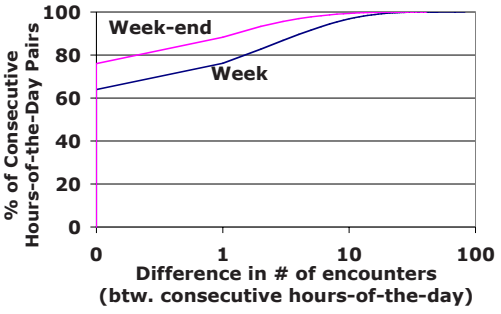

Fig. 5. CDF of the differences of each individual's number of encounters for the same hour-of-the-day for week days and week-end days. People's encounter rates are predictable. An individual's number of encounters per hour remains the same 64% of the time Monday through Friday and 76% of the time on Saturdays and Sundays.

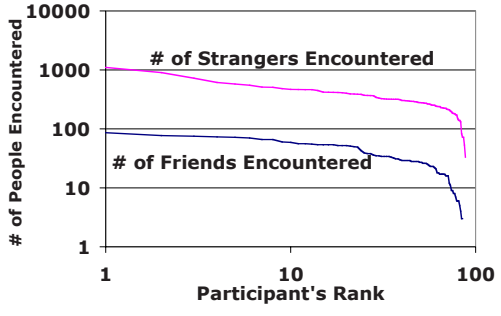

Fig. 6. Distribution of the number of friends and the number of strangers for each participant, on a log-log scale. These curves appear to follow a power-law distribution, suggesting that the friend and the stranger network graphs are scale-free. Many social networks have been previously found to be scale-free [2].

individual's encounter rate remains the same during two consecutive hour slots 64% of the time Monday through Friday and 76% of the time on Saturdays and Sundays. Also, the changes in the rate between consecutive hour slots are very small; this rate changes by more than 5 encounters less than 7% of the time. These results show that people's encounter rates are very predictable during the same hour of the day.

2.5 The Friend and the Stranger Networks Are Scale-Free

Many social networks have been previously found to be scale-free [2]. One of the distinguishing characteristics of scale-free networks is that their node degree distribution follows a power-law relationship $P(k) = k^{-\gamma}$. In power-law networks, a small number of nodes are highly connected, while most nodes have low connectivity.

Figure 6 shows the distributions of the number of friends and the number of strangers encountered by the 88 participants on a log-log scale. Both curves appear to follow a

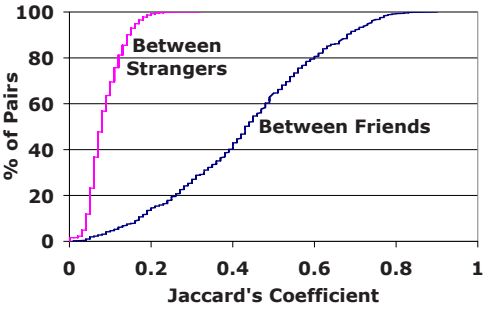

Fig. 7. Distribution of Jaccard's coefficients between the neighbor sets of the friend and the stranger networks, respectively. We use the Jaccard's coefficient to measure the similarity between the encounter sets of two people. The data shows that the friend network has more similar neighbor sets than the stranger network. This suggests that the friend network is more clustered than the stranger network.

similar power-law distribution for most participants (a power-law distribution appears as a straight line on a log-log plot). We further examined these curves' tails since they do not seem to follow a power-law distribution. We found that many of these participants are not fully active over the entire trace duration; we believe that their lack of activity makes them encounter fewer friends and fewer strangers, respectively.

2.6 The Friend Network Has High Local Clustering

Many social networks have been shown to have a high local clustering coefficient [2]. In this section, we examine whether the friend and the stranger networks are highly clustered.

Unfortunately, the trace methodology prevents us from measuring the clustering coefficient in both the friend and the stranger networks. While we have full information about participants, we lack complete information about their friends or their strangers. Instead, we measure the similarity of the participants' neighbor sets in these two networks. We use the Jaccard's coefficient to measure similarity as a first order approximation of the degree of clustering present in these networks. The Jaccard's similarity coefficient of two sets is the size of their intersection divided by the size of their union – $J(A, B) = |A \cap B|/|A \cup B|$. Two identical sets have a Jaccard's coefficient of 1, and two completely disjoint sets have a coefficient of 0.

Figure 7 shows the distribution of the Jaccard's coefficient for all pairs of friends and strangers in our data. The data suggests that there is a substantial difference between these two networks. In the friend network, the neighbor sets appear similar, with a median Jaccard's coefficient of 0.43, over five times higher than the median Jaccard's coefficient of the stranger network (0.08). In the friends graph, over 90% of all pairs have more similar neighbor sets than almost all (95%) pairs of strangers.

2.7 Summary

This section used trace data to identify key properties of people encounters. From this data, we find several important observations:

- While most pairs of people encounter sporadically, most encounters are generated by pairs of people encountering often. This suggests the presence of two types of encounters in the data: encounters between friends and encounters between strangers.
- People encounters are driven by diurnal and weekly cycles. Once we account for time-of-day and day-of-the-week effects, the number of encounters of an average person is consistent. People's encounter rates are predictable during the same hour of the day for week days and week-end days.
- Both the friend and the stranger graphs are scale-free. The node degree distribution in these networks follow a power-law distribution, suggesting that while few nodes have many friends (or strangers), most nodes have few friends (or strangers, respectively).
- In the friend network, the participants' neighbor sets are similar, where in the stranger network, they are not. This suggests that the friend network has a high degree of clustering.

3 A Social Networking-Based Simulator of People's Encounters

The premise of our work is that the performance of mobile applications and protocols can improve if they incorporate information about people's social relations. This section presents a simulator of a mobile environment that enables us to explore our premise. Our simulator captures key social and temporal aspects of mobile environments, such as friend encounters, stranger encounters, and how the number of encounters varies with the time-of-the-day and the day-of-the-week. From these parameters, it produces a large-scale synthetic trace of people encounters over time.

3.1 Simulator Description

As previously discussed, a person's friend encounters are different from their stranger encounters in important ways. To capture this distinction, our simulator uses two different models to generate friend and stranger encounters. We use the Watts-Strogatz small-world model [33] when generating encounters between friends, while we use a version of the Barabasi scale-free model [3] when generating encounters between strangers.

The Watts-Strogatz small-world model captures the high clustering property specific to the friend social networks. A clustered friend graph preserves the transitive nature of friendships: an individual's friends must be related to each other in a realistic manner. Our simulator captures this transitive nature of friendships: if A and B are friends, and B and C are friends, then the probability of A and C being friends is higher than a random chance. This transitivity property of friendships is important to the flow of information in social networks [9].

Table 2. Simulator structure and notation. These parameters' settings reflect the values seen in the trace we analyzed.

Symbol	*Meaning*	**Base value**
N	*# of nodes*	20,000
f	*# of friends per node (Watts-Strogatz)*	20
α	*Zipf parameter for stranger encounters' distribution*	1.129
p	*probability of encountering a friend*	63.1%
$\lambda_{week\ day}$	*hourly rate of encounters (vector with 24 values one for each hour of a week day)*	(0.1, 0.06, 0.06, 0.04, 0.05, 0.06, 0.03, 0.02, 0.07, 0.5, 1.03, 0.97, 1.58, 1.37, 1.52, 1.73, 1.76, 1.37, 1.62, 0.76, 0.46, 0.37, 0.24, 0.15)
$\lambda_{week\text{-}end\ day}$	*hourly rate of encounters (vector with 24 values one for each hour of a week-end day)*	(0.15, 0.15, 0.09, 0.03, 0.02, 0.03, 0.02, 0.02, 0.05, 0.06, 0.08, 0.16, 0.19, 0.3, 0.34, 0.34, 0.33, 0.31, 0.19, 0.29, 0.26, 0.25, 0.2, 0.18)

The Watts-Strogatz model places N nodes on a ring and connects each with K of its neighbors ($K/2$ on each side). To randomize the graph, each edge is rewired to a random node with a small probability. The resulting graph has a small average path length and a high clustering coefficient relative to a completely random graph with the same number of nodes and edges, as desired [33]. When the simulator generates a friend encounter, it selects a node at random and then selects another node at random from the first node's set of friends. An encounter will then be generated between these two nodes. Each node's friend set remains fixed over the course of the simulation, since the friend network is not altered once the simulation begins to run.

This model's main limitation is that the nodes' degree distribution is not a power-law, but more similar to that of a regular graph. Several extensions to this model address this limitation [15,11,8]; we plan to examine more sophisticated small-world models in future work. However, since friends on average compose less than 7% of each individual's unique contacts, the overall degree distribution of the encounter network is driven almost entirely by stranger encounters.

We generate stranger encounters using an approach inspired by the preferential attachment model proposed by Barabasi et al. [3]. Barabasi's model grows a scale-free network by adding one node at a time. Each new node attaches itself to a fixed number of existing nodes with a probability proportional to each existing node's degree. Although each node enters the network with a fixed number of edges, the node may acquire additional edges as new nodes link to it when they are added to the network. One side-effect of Barabasi's model is that the last opportunity for two nodes to be linked by an edge is when the second node of the pair is added to the network. Once added without a link between them, two existing nodes can never encounter each other.

Our simulator makes a small modification to this model. Instead of growing the network one node at a time, it assumes a closed population. Each node is pre-assigned a Zipf-based popularity score that determines the probability of selecting this device when generating stranger encounters. The Zipf law is a type of a power-law commonly found in nature. To generate a stranger encounter, the simulator randomly selects two nodes with a probability proportional to their respective Zipf scores. An encounter will then be generated between these two devices. The simulator is careful not to pick a pair of friends when generating a stranger encounter.

Our method of generating stranger encounters ensures that at any time, the probability of two nodes meeting each other in a stranger encounter is non-zero, except

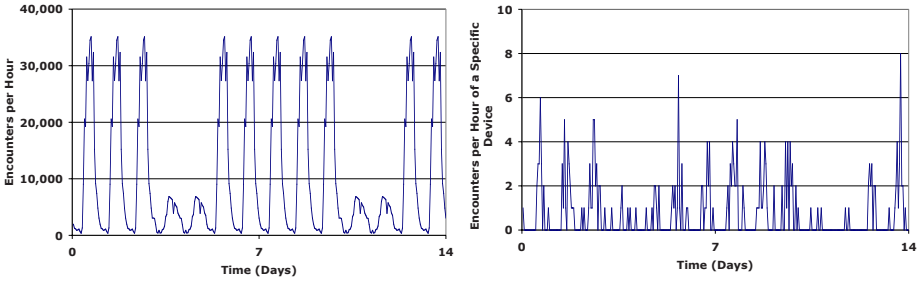

Fig. 8. Encounters produced by our simulator. The number of encounters per hour for all 20,000 people on the left, and the number of encounters for one specific person on the right. Since the number of encounters per hour is fixed (based on the hour of the day), each week day and each week-end day appear indistinguishable on the left. However, individual persons do not have cyclical behaviors. On the right, we show how an average person's number of encounters per hour varies.

when the two nodes are friends. While in the long-run this violates the power-law property of nodes' degrees, we believe that it captures adequately the behavior of a closed population: in a fixed set of people, everybody eventually meets everybody else. However, we never experience this saturation regime in any of our simulations.

Table 2 summarizes the parameters used in our simulations. We use our simulator to generate a two week synthetic trace of encounters. We chose parameter values from a two week period of the MIT Reality Mining trace. We do not simulate the encounters' durations and we assume a fixed number of people in the system. Our simulator generates requests as follows. On average, 63.1% of a person's daily encounters are with friends and 36.9% with strangers. To generate an encounter, our simulator creates a friend encounter with probability 0.631 and a stranger encounter with probability 0.369. We hypothesize that the underlying stranger popularity is driven by Zipf's law. We estimated the Zipf's parameter from a two week portion the trace to be $\alpha = 1.129$. The encounter rates vary according to the time-of-the-day and the day-of-the-week. Since the number of encounters remains constant on an hourly basis, we use 24 hourly rates during a week day and another 24 hourly rates during a week-end day.

Despite our ability to estimate many of the input parameters from the trace data, it is not possible to directly estimate N (the number of people) with any confidence. For that reason, we leave N as a free parameter, adjusting it to obtain as tight a correspondence between the simulator and the data trace as possible. Figure 8 illustrates the encounter patterns captured by our simulator.

3.2 Simulator Validation

Our simulator's main goal was to capture the specific characteristics of friend and stranger encounters. Many of these properties are built-in: the rate of encounters, the fraction of friend versus stranger encounters, the heavy-tailed distribution of friend and stranger popularities, and the heavy clustering of the friend network. We validated our model by measuring the speed of information propagation in our synthetic trace and

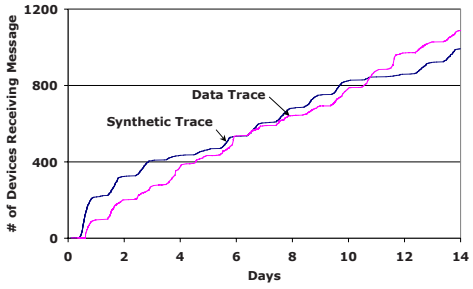

Fig. 9. Predicted versus measured flow of information in a restricted trace. The speed of information flow in the network is reflected in the curves' slopes. The synthetic curve's characteristics are close to the real data trace's characteristics.

comparing it to the data trace. The data trace is restricted; it does not capture encounters between non-participants. In contrast, our synthetic trace captures all encounters between all people. To match the data trace's environment, we selected a set of nodes from our synthetic trace to serve as our instrumented participants. We matched the number of participants selected to the two week Reality Mining trace we used to parameterize our model. We did not choose the participants randomly. Instead, we chose a subgraph in the friend network and we marked all nodes as participants in our validation experiment. In this way we ensured that participants have strong friendship ties among them, similar to the the data trace's participants, who come from a single environment and are likely to be socially related.

Next, we removed all encounters between unselected nodes in our synthetic trace since these correspond to encounters between non-participants. Thus, we were able to produce a synthetic trace with an experimental restriction similar to the original trace. We used the number of encounters in our restricted synthetic trace to calibrate how to scale up the rate of encounters in our simulator. Initially, we scaled up the rate of encounters linearly with the size of the population. However, this led to an unrealistically high number of encounters. Instead, we calibrated the scaling factor so that the number of encounters in the restricted synthetic trace matches the number of encounters in the real trace. The same scaling factor also led to an accurate distribution of encounters between participant-to-participant and participant-to-non-participant encounters.

Figure 9 shows how information propagates through our restricted synthetic trace and through the original trace. For this, we simulated how a message sent by a random participant spreads through the network over time. When the total number of people in the simulation (N) is set to 20,000, the rate of information propagation in the synthetic network is close to the real trace.

4 Exploiting Social Interactions in Mobile Systems

In this section, we use our social networking-based simulator to investigate the potential benefits of using social networking information to three mobile systems: (1) the performance of DTN routing protocols, (2) slowing down the propagation of mobile worms,

and (3) improving the query hit rate of a mobile file-sharing application. We examine each of these applications in turn.

4.1 Routing in Delay Tolerant Networks (DTN)

In this section, we examine the performance of DTN routing protocols from a social networking perspective. After presenting a brief primer on DTN routing protocols, we study their performance in the presence and in the absence of social information. Our findings will show that, by using social information, routing protocols can achieve substantial performance gains.

A Brief Primer on DTN Routing

Various DTN routing protocols make different assumptions about the knowledge available to network nodes. While some assume that nodes have no knowledge about the state of the network, others assume that nodes have access to different types of information, such as the topology of the network, the average time between successive encounters of two nodes, who the congested nodes are, or the network traffic matrix [14].

Most protocols assuming no knowledge about the network are based on epidemic routing [14,31,26]. These algorithms are optimal – they *always* deliver the message over the *shortest* available path. They are also well-understood and relatively easy to implement and deploy. Although optimal, epidemic routing is expensive and unscalable since a message can potentially reach all nodes in the network.

To control the flooding of packets, epidemic protocols typically associate a time-to-live field with each packet or they restrict their forwarding decisions. For example, in the First Contact protocol [14], a node only forwards along the first available link. While these techniques reduce the cost of epidemic routing, they also reduce the protocols' performance, and they sometimes fail to deliver the packet. In fact, the First Contact protocol has been known to perform poorly in general since the chosen next-hop is essentially random [14]. In summary, the DTN routing protocols that assume no knowledge about the network perform poorly: they are either unscalable in practice (uncontrolled epidemic routing) or their delivery success rates are low (first contact routing) [16].

Other DTN routing protocols assume some knowledge about the state of the network [19,14,16,30]. All these protocols try to compute shortest paths to the destination assuming that certain network information is available. Some assume little extra information, such as the average waiting time until the next contact for an edge, while others assume that all nodes know the entire network topology at all times. The performance of these DTN routing protocols varies depending on the amount of information available and the network dynamics. A comprehensive evaluation of these protocols for several DTN scenarios is presented in [14].

Incorporating Social Networking in DTN Routing

Social information is another type of information that is often readily available to nodes in a DTN scenario. This information can help DTN routing protocols make more informed decisions to whom to forward a specific message. For example, when routing

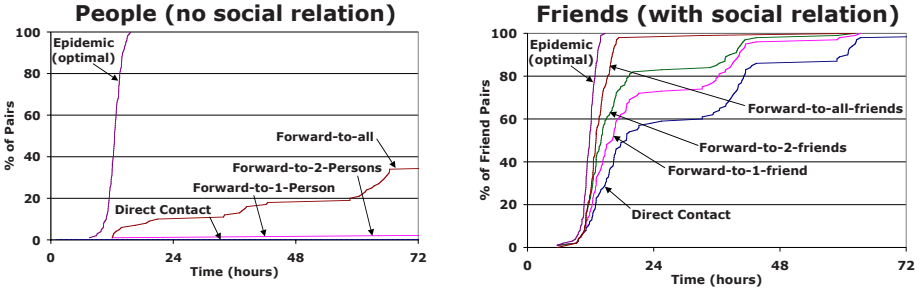

Fig. 10. The performance of DTN routing protocols. In the "direct contact" protocol, the sender does not forward the message to any intermediary; instead it waits to encounter the destination. In "forward-to-k-persons/friends", the sender forwards the message to the first k persons (or friends). The sender and the intermediaries do not subsequently forward the message unless they encounter the destination. "Forward-to-all" forwards to all persons encountered by the sender. Epidemic routing floods the message to all nodes. On the left, the distribution of a message's delivery times between 100 pairs of random people is shown. On the right, the same distribution between 100 pairs of friends is shown; in this experiment, all forwarding decisions are restricted to friends only. The routing protocols perform significantly better in the presence of social information.

between friends, a protocol could prefer selecting intermediaries who are friends with either the source or the destination. Friends are more likely to be clustered and to encounter one another. To quantify the performance of incorporating social networking in DTN routing, we used our simulator to evaluate several protocols in the presence and in the absence of social information.

While we evaluated a suite of DTN protocols, in this paper, we present only four protocols: "direct contact", "forward-to-1-person", "forward-to-2-persons", and "forward-to-all" [32]. In "direct contact", the sender does not forward the message to any intermediary; instead it waits to encounter the destination. In "forward-to-1-person", the sender forwards the message only to the first person encountered. There is no subsequent forwarding; the message is delivered only when the sender or the intermediary encounters the destination. The "forward-to-2-persons" works similarly, the sender forwarding to the first two persons encountered. Finally, in "forward-to-all" the sender forwards the message to all persons it encounters. Note that this is different than epidemic routing, since in "forward-to-all", none of the intermediaries forward to any nodes other than the destination. We also implemented the optimal, epidemic routing protocol to serve as a baseline of comparison.

On the left, Figure 10 shows the distribution of delivery times of 100 messages sent between 100 pairs of people randomly chosen. With epidemic routing, all messages are successfully routed in less than 16 hours. However, the cost of epidemic routing is immense: over half a million messages are being forwarded throughout the network. On the other hand, the other four DTN routing protocols perform very poorly. In two weeks, "direct contact" is unable to deliver even one single message.

On the right, Figure 10 shows how these routing protocols perform in the presence of social information. For this, the simulator selected 100 random pairs of friends and it restricted all the protocols to only forward to a friend of the source or the destination. To capture the optimal delivery times, we left the epidemic routing protocol to forward to any person. As Figure 10 shows, "direct contact" delivers 50% of the messages in less than 19 hours, taking only an extra 7 hours over the optimal epidemic routing protocol. Forwarding to one friend reduces the delivery times of half of the messages by two hours and 45 minutes, and forwarding to two friends adds an additional two hours of savings to the delivery times. By forwarding the message to all friends of the source or the destination, 98% of all messages are delivered in less than 17.5 hours. These routing protocols' performance is close to optimal without the huge overhead of flooding the entire network – each message is forwarded a small number of times only, at most on the order of the number of friends of the source and the destination. We also evaluated these protocols when routing between people with no social relation and forwarding to the source or the destination's friends; the protocols' performance is much more modest.

In summary, our findings show that social information leads to substantial performance gains for DTN routing protocols. While our experiments only separated friend from stranger encounters, we believe that a more refined treatment of social information (e.g., identifying social groups and social behavior) is likely to further improve these protocols' performance. We plan to investigate this in future work.

4.2 Slowing the Spread of Worms

In this section, we examine whether firewalls that discriminate between traffic sent by friends and traffic sent by strangers can slow down the propagation of a worm in a mobile network. We use the propagation speed of a worm infection as a lens to measure the effectiveness of firewall rules based on social networking.

The research community has already started to investigate the feasibility and the propagation dynamics of worms in mobile networks [5,6,34,29]. While no large-scale mobile worm outbreak has been reported so far, several reports of worms spreading over the Bluetooth protocol in a cell-phone environment exist [7,12]. The consequences of a malicious program infecting a large number of cell-phones can be disastrous. For example, such a worm could launch a DoS attack by overloading a segment of the cellular network. Similarly, a spyware program infecting cell-phones could collect personal information. By slowing the propagation of a worm in a mobile network, security experts can have more time to create and distribute a software patch repairing the vulnerability exploited by the worm.

An effective way of slowing the propagation of a worm is to firewall devices to prevent them from receiving traffic from all other devices. While such a measure would be very effective, this solution is also unappealing – it will prevent devices from using their radio interfaces for legitimate applications. Instead, a firewall that allows traffic only from a select set of devices could greatly slow the spread of a worm but allow many applications to function normally. For example, a firewall that accepts traffic only from friends would not prevent people from using their devices to exchange data with people they know. In this way, several applications, such as exchanging chat messages or files with friends, can still function in the presence of such firewalls.

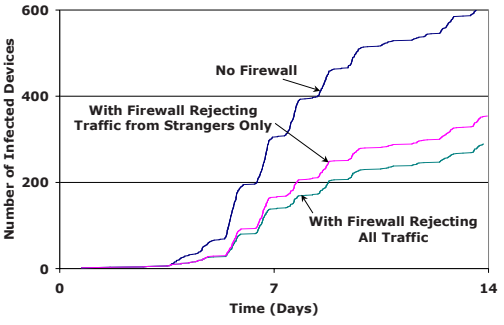

Fig. 11. The propagation of a mobile worm over time. In this experiment, 5% of devices (out of 20,000) are vulnerable. The rate of infection is presented when no firewalls are present in the system and when 30% of vulnerable nodes (1.5% of the entire population) are firewalled. We show the results when running a firewall rejecting all traffic and when running a firewall rejecting traffic from strangers only. The two firewalls are almost as effective suggesting that social based firewalls can provide a good compromise between preventing a worm from infecting devices and allowing some network applications to still function.

We use our simulator to investigate the effectiveness of such firewalls in a mobile network. In our experiments, a worm outbreak occurs by initially infecting one randomly chosen node. We randomly select 5% of the population to be vulnerable; our fraction of vulnerable devices is low since the most virulent known worms, such as Internet worms, only infected a relatively small fraction of all Internet nodes [25,24]. We select 30% of the vulnerable devices (1.5% of the entire population) to be equipped with a social networking firewall. We measure the number of infected devices with and without social networking firewalls.

Figure 11 shows our results. Without a firewall, a worm can infect half of the vulnerable devices in 9.5 days. While the worm does not propagate very quickly for the first five days, over 30% of vulnerable devices are infected in one week. The rate of propagation is also influenced by the network's temporal properties – the worm "slows-down" during nighttime, but it then resumes a quick infection pace on the next day.

Even when a small fraction of devices (1.5%) turn on a social networking firewall, the worm infection slows down significantly. Only a small fraction of vulnerable devices (10%) are infected in the first week of the outbreak. It takes over two weeks to infect half of the vulnerable devices, a delay of over five days when compared to the time it takes to infect half of the population in the absence of such firewalls. The effectiveness of the social networking firewall is almost close to optimal – a perfect firewall would only prevent an additional 27 devices from becoming infected in one week.

These results suggest that social networking firewalls can slow down the spread of a worm allowing for extra time to distribute a patch to the uninfected but still vulnerable devices. At the same time, devices running such firewalls can continue to use the network to communicate with their friends. These findings show that social networking firewalls can provide an attractive solution to both users and security experts in the face of a large-scale worm outbreak.

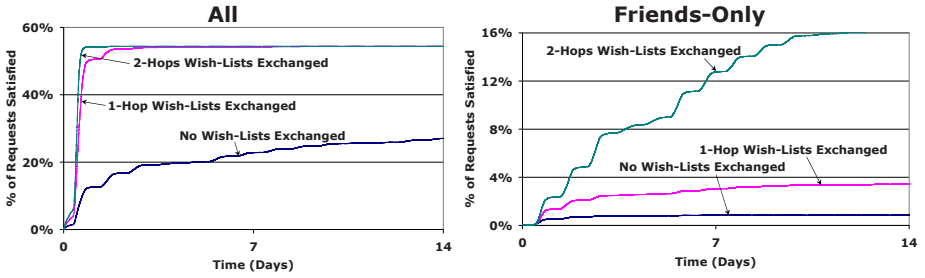

Fig. 12. The fraction of successful requests over time in mobile P2P systems. We implemented three file exchange policies: "no wish-lists exchanged", "1-hop wish-lists exchanged", and "2-hops wish-lists exchanged". A peer downloads a file if either it wants it or it has previously received a wish-list containing this file. On the left, content is exchanged between all peers. On the right, content and wish-lists are exchanged between friends only. When restricting content exchanges to friend encounters only, the rate of successful requests decreases drastically.

4.3 File-Sharing in Mobile P2P Systems

Recently, several companies have started to offer file-sharing software for mobile phones that allow users to share ring-tones, music, games, photos, and video [28,17]. In mobile P2P systems, content exchange is driven by the users' social interactions – people encounter each other in social settings and they use their cell-phones to exchange content. To understand these systems' behavior, we need to understand to what extent content propagation is driven by friend versus stranger encounters. In this section, we examine the performance of several file exchange protocols in a mobile P2P file-sharing system from a social networking perspective.

P2P systems must provide incentives for participants to upload and share content. In the absence of such incentives, many peers offer little or no data to the system. Such peers are known as "free-riders" [1]. Creating a suitable incentive mechanism in a P2P system and enforcing it in a decentralized manner is a challenging problem and an active area of research [21,20,27]. On the other hand, much of the content exchange in a mobile network occurs in social settings: friends share content among themselves. Such environments offer a natural set of incentives: friends are likely to share data or even forward data on each other's behalf. If exchanging content between friends, without involving strangers, can satisfy most people's requests, the need for an explicit incentive mechanism design is greatly diminished.

To examine whether content exchange is driven by friend encounters or by stranger encounters in a mobile P2P system, we performed the following experiment. We started with a trace of P2P file exchanges in Kazaa, a popular Internet P2P system, collected at the University of Washington [10]. Each of the 24,578 nodes in this trace has a "wish-list" and a "have-list". The wish-list corresponds to all of the files that the node downloads from its peers over the course of the trace, while the have-list is the set of all files that this node is willing to provide to its peers. From this trace, we selected 20,000 peers and we mapped them to the 20,000 people whose encounters are generated by our simulator. The mapping is done according to peers' popularities: the peer having

the largest have-list is mapped to the participant with the highest number of encounters in our simulator. When two peers encounter, a file-exchange policy dictates which files and wish-lists the peers should exchange. Since our simulator does not capture contact durations, we assume that file transfers occur instantaneously.

We implemented three file-exchange policies by varying the number of hops wish-lists are exchanged in the network. In the first policy, "no wish-lists exchanged", a content exchange occurs only if one peer wants a piece of content present on the other peer. No content is downloaded on behalf of others. In the "1-hop wish-lists exchanged" policy, wish-lists are exchanged between neighbors only (wish-lists are flooded with a time-to-live (TTL) of 1.) A peer downloads a file if either it wants it, or it has previously received a wish-list containing this file. In this way, content is replicated on peers who have previously encountered someone wanting the file. The "2-hops wish-lists exchanged" policy behaves similarly, except the wish-lists' TTL is set to 2.

To evaluate whether peers can find content among their friends, we conducted two sets of experiments: one in which all peers share content among themselves, and one in which content sharing is restricted to friend encounters only. Figure 12 shows our findings. On the left, we show the fraction of requests satisfied over time when all peers exchange content. In two weeks, only 27% of requests are satisfied when no wish-lists are exchanged. On the other hand, if wish-lists are exchanged between neighbors, 54% of requests are satisfied. Exchanging wish-lists between peers can substantially improve the users' query hit rate in the system.

On the right, Figure 12 shows the fraction of requests satisfied over time when only friends exchange content. In two weeks, less than 1% of requests are satisfied when wish-lists are not exchanged. Even if wish-lists are exchanged along two hops, only 15% of requests are satisfied over two weeks. These findings suggest that restricting content exchange only to friend encounters drastically reduces the rate of successful requests. In our experiments, peers find three times fewer files when restricting their content exchange to friend encounters only.

Our findings illustrate that mobile P2P systems cannot rely on friend encounters to deliver content to their users. Although such a scheme could provide a natural set of incentives to a system, it would significantly penalize the users' query hit rate. Instead, like the file-sharing systems present on the Internet, P2P systems in mobile environments must rely on developing alternate incentive schemes to ensure that peers contribute their content.

5 Conclusions

In this paper we used social networking-based simulations to show how three mobile systems can exploit people's social relations to improve performance and query hit rate. We first showed that simple DTN routing protocols that avoid forwarding to strangers work very well when routing between friends. Next, we found that firewalls allowing traffic from friends while rejecting traffic from strangers are effective at slowing down the spread of worms in mobile environments. Finally, we showed that mobile P2P file-sharing systems must rely on strangers to exchange content to satisfy their users' requests.

Acknowledgments. We would like to thank Mostafa Ammar for his encouragement to pursue our ideas. We gratefully acknowledge the use of Bluetooth data from Nathan Eagle at MIT. Finally, we wish to thank the anonymous reviewers for their comments and feedback.

References

1. Adar, E., Huberman, B.: Free riding on Gnutella. First Monday 5(10) (October 2000)
2. Albert, R., Barabasi, A.-L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74(1), 47–97 (2002)
3. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
4. CNET News.com. Mobile browsing becomes mainstream (2006), http://news.com.com/Mobile+browsing+becoming+mainstream/2100-1039_3-606 2365.html
5. Cole, R.G.: Initial Studies on Worm Propagation in MANETS for Future Army Combat Systems (2004), http://stinet.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&iden tifier=ADA431999
6. Cole, R.G., Phamdo, N., Rajab, M.A., Terzis, A.: Requirements of Worm Mitigation Technologies in MANETS. In: Principles of Advanced and Distribution Simulation (2005)
7. ComputerWorld. Cabir Worm Wriggles into U.S. Mobile Phones (2005), http://www.computerworld.com/securitytopics/security/virus/story/0,10801,999 35,00.html
8. Ebel, H., Davidsen, J., Bornholdt, S.: Dynamics of social networks. Complexity 8(2), 24–27 (2002)
9. Granovetter, M.S.: The strength of weak ties. The American Journal of Sociology 78(6), 1360–1380 (1973)
10. Gummadi, K.P., Dunn, R.J., Saroiu, S., Gribble, S.D., Levy, H.M., Zahorjan, J.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: 19th ACM Symposium on Operating Systems Principles (SOSP), Bolton Landing, NY, USA, October 2003, ACM Press, New York (2003)
11. Holme, P., Kim, B.J.: Growing scale-free networks with tunable clustering. Physical Review E 65(026107), 1–4 (2002)
12. InfoSyncWorld. First Symbian OS Virus to Replicate over MMS Appears (2005), http://www.infosyncworld.com/news/n/5835.html
13. InfoWorld: More mobile Internet users than wired in Japan (July 2006), http://www.infoworld.com/article/06/07/05/HNjapannetusers_1.html
14. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: Proceedings of ACM Sigcomm, Portland, OR, USA (2004)
15. Jin, E.M., Girvan, M., Newman, M.E.J.: The structure of growing social networks. Physical Review E 64(046132), 1–8 (2001)
16. Jones, E.P., Li, L., Ward, P.A.S.: Practical routing in delay-tolerant networks. In: Proc. of ACM Sigcomm Workshop on Delay-Tolerant Networking, Philadelphia, PA, USA (2005)
17. JuiceCaster. Share your mobile life with juicecaster (2007), http://www.juicecaster.com
18. Kangourouge. Proxidating, the first ever Bluetooth dating software for mobile phones (2007), http://www.proxidating.com
19. Lindgren, A., Doria, A., Shelen, O.: Probabilistic routing in intermittenly connected networks. In: Proceedings of ACM Mobihoc, Annapolis, MD, USA (2003)

20. Liogkas, N., Nelson, R., Kohler, E., Zhang, L.: Exploiting bittorrent for fun (but not profit). In: Proceedings of Proceedings of 5th International Workshop on Peer-to-Peer Systems (IPTPS), Santa Barbara, CA, USA (2006)
21. Locher, T., Moor, P., Schmid, S., Wattenhofer, R.: Free riding in bittorrent is cheap. In: Proceedings of HotNets, Irvine, CA, USA (2006)
22. Milgram, S.: The Familiar Stranger: An Aspect of Urban Anonymity. Addison-Wesley, Reading (1977)
23. MIT Media Lab: Reality Mining. `http://reality.media.mit.edu/`
24. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., Weaver, N.: The Spread of the Sapphire/Slammer Worm. Technical Report CAIDA, ICSI, Sillicon Defense, UC Berkeley EECS and UC San Diego (January 2003)
25. Moore, D., Shannon, C., Brown, J.: Code-red: a case study on the spread and victims of an internet worm. In: 2002 Internet Measurement Workshop (November 2002)
26. Niculescu, D., Nath, B.: Trajectory based forwarding and its applications. In: Proceedings of Mobicom, San Diego, CA, USA (2003)
27. Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A.: Do incentives build robustness in bittorrent. In: Proceedings of 4th Usenix Symposium on Networked Systems Design and Implementation (NSDI), Cambridge, MA, USA (2007)
28. Pogo. Pogo browser (2007), `http://www.pogo42030.co.za`
29. Su, J., Chan, K.K.W., Miklas, A.G., Po, K., Akhavan, A., Saroiu, S., de Lara, E., Goel, A.: A preliminary investigation of worm infections in a bluetooth environment. In: 4th Workshop of Recurring Malcode (WORM), Fairfax, VA, USA (2006)
30. Su, J., Goel, A., de Lara, E.: An empirical evaluation of the student-net delay tolerant network. In: 3rd International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS), San Jose, CA, USA (2006)
31. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks. Technical Report CS-200006, Department of Computer Science, Duke University (April 2000)
32. Wang, Y., Jain, S., Martonosi, M., Fall, K.: Erasure-coding based routing for opportunistic networks. In: WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, pp. 229–236. ACM Press, New York (2005)
33. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
34. Yan, G., Eidenbenz, S.: Bluetooth worms: Models, dynamics, and defense implications. In: 22nd Annual Computer Security Applications Conference, Miami Beach, FL, USA (2006)