Hearsay: Suppressing spam using trust in mobile social gossiping networks

Master's Thesis in Computer Science

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Onno Steenbergen
o.steenbergen@student.tudelft.nl

4th June 2012

Author
Onno Steenbergen (o.steenbergen@student.tudelft.nl)
Title
Hearsay: Suppressing spam using trust in mobile social gossiping networks
MSc presentation
30th May 2012

Graduation Committee

Prof. Dr. K.G. Langendoen	Delft University of Technology
Dr. E. Onur	Delft University of Technology
Prof. Dr. Ir. D.H.J. Epema	Delft University of Technology
Ir. N. Brouwers	Delft University of Technology

Abstract

The development of social networks has changed the way we communicate from a personal conversation into a broadcast with a world-wide audience. Social networks have proven themselves useful during protests and natural disasters by providing a platform to share ideas and offer help to those in need. However, due to the reliance on the Internet there have been situations where social networks were unable to function. Examples are broken submarine cables and countries that have restricted Internet access for residents, such as Egypt and Libya during the uprising in the beginning of 2011. Alternative communication methods are available, such as mobile ad-hoc networking, but suffer a common problem, large quantities unwanted messages, better known as *spam*.

We propose a novel approach called "Hearsay" to combat spam. Our approach suppresses spam in gossiping networks by utilising social network information. Gossiping networks are characterised by users carrying devices capable of creating ad-hoc communication links with devices nearby. Messages are exchanged using a gossip protocol, spreading them throughout the network. Without suppression a message would reach every user making it ideal for spamming. Prioritising messages based on personal and social relations keeps messages within a social group. Users assign a rank to each user they interact with and based on that rank messages are forwarded to other members.

Through simulations we show the feasibility of Hearsay, as a large-scale deployment is not possible within the set time frame. We ran our simulations within the multi-agent modeling environment NetLogo, which showed the effectiveness of using trust to suppress spam.

Preface

This thesis is the result of the work I did for more than a year. The original subject was indoor localisation using mobile devices and it gradually changed to battery efficiency of mobile communication technology. During this period new mobile phones were bought that included an ANT communication chip. Although this chip was included to communicate with sensors it was possible to send messages between two mobile phones. A proof-of-concept was built to demonstrate a message service. Although the technology had its limitations it sparked my interest in building a communication network between mobile phones.

My acknowledgements go to Kavitha Muthukrishnan for accepting me as her master student. Niels Brouwers, my daily supervisor after Kavitha left, for showing me how to conquer the world and Koen Langendoen for introducing me to the topic of Wireless Sensor Networks. Finally I would like to thank my friends, family and colleagues for their support.

Onno Steenbergen

Delft, The Netherlands
4th June 2012

Contents

Chapter 1

Introduction

At the beginning of the year 2011 the world media was focused on the social uprising in African and Arabic countries [11]. With astonishing speed the revolt spread from Tunisia to Egypt, Morocco, Algeria, Yemen, Oman, Bahrain, Syria, Iran, Lebanon, Saudi Arabia and Libya.

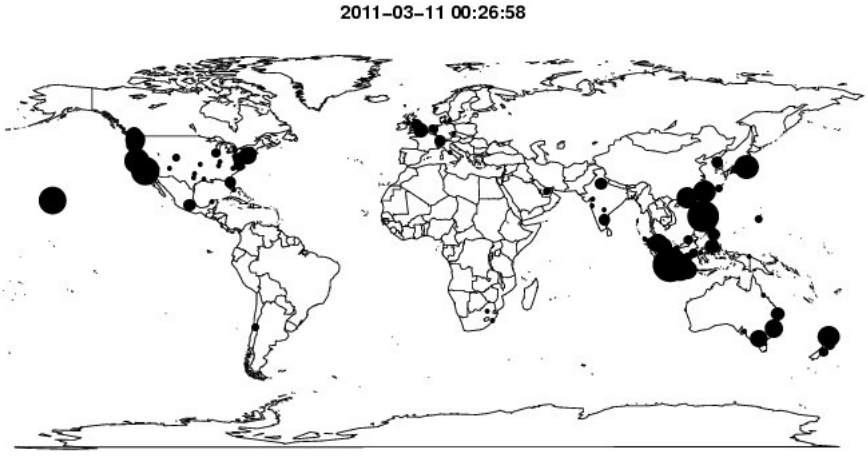
Social media, such as YouTube[1], Twitter[2] and Facebook[3], played a crucial role in coordinating the uprising. Governments in Egypt and Libya tried to limit Internet access but it was already too late [9, 14]. The so called 'rebel fighters' used their improvisational weapons to gain control of the country, while social media was used to win the hearts and minds of the public.

This event was followed by an earthquake in Japan where social media again played an important role, from raising awareness and donations to giving out locations of shelters and food supplies. Social networks handled thousands of messages per second to support Japan and its people [10, 39]. Figure 1.1 shows how users on Facebook, the largest social network [1], mentioned the earthquake in their status updates.

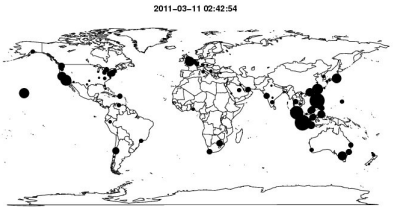
Social media has become a key component for communication, but is has a critical dependency on the availability of the Internet. Although in Japan the residents could still use the Internet there were other events where the service was limited. One example is the submarine cable disruption caused by anchors in 2008 [58]. Multiple optical cables were damaged causing Internet disruptions in the Middle East and Asia. An other example is censorship, where governments restrict residents to visit certain sites or use some communication techniques [22, 45].

When the Internet is not available setting up new communication links takes time, money, and other resources, such as a repair ship for submarine cables. During this period of disruption a basic communication network can replace the Internet, whether it is to survive a disaster or avoid censorship.

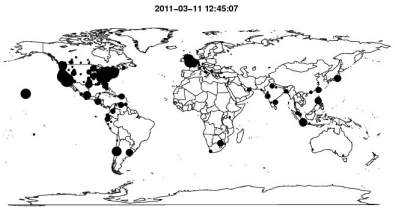
[1] `www.youtube.com`
[2] `www.twitter.com`
[3] `www.facebook.com`



(a) 11-03-2011 00:26:58



(b) 11-03-2011 02:42:54



(c) 11-03-2011 12:45:07

Figure 1.1: A visualisation of messages mentioning the earthquake in Japan. In 1.1(a) most messages are from people who felt the earthquake. 1.1(b) and 1.1(c) how activity changes from concerned citizens of Japan to the people in United Stated and Europe who just heard the news.

Alternative methods of communication can be found in new mobile technologies. Mobile phones have become ubiquitous; the Facebook statistics show [1] that a third of the social network users already use their mobile phone to interact with other users. These 250 million members hold a device capable of creating an *ad-hoc network* between users. Wi-Fi Direct, Bluetooth and other direct communication methods allow users to directly exchange messages between devices without the need for an intermediary service provider. In such a network one can *gossip* directly to any phone that is close enough, similar to *word-of-mouth* or viral communication. Consider a service like Twitter, where messages are broadcasted to a set of interested parties [27]. The message is copied *opportunistically*, without user interaction, from one phone to another.

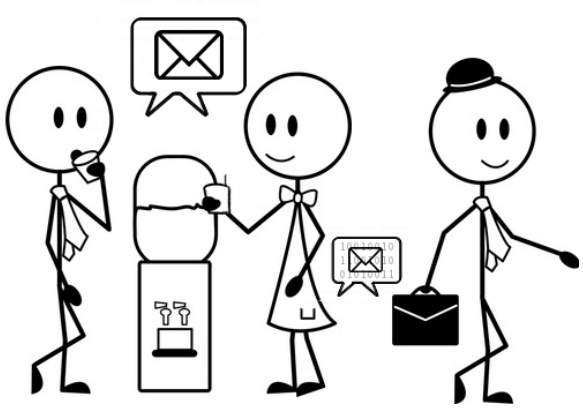


Figure 1.2: Word-of-mouth and digital gossiping

Like the gossiping co-workers in Figure 1.2, mobile phones can exchange messages to any other device in range. Each device spreads received information to other devices. As a result all devices quickly receive the rumour and it becomes important to distinguish between fact and fiction. People can judge a rumour and dismiss it but a computer can not do this easily. Therefore the act of deliberately spreading false information, so-called *disinformation*, is a well-known problem in gossiping networks [19].

The reason to disinform members of the social network can be diverse, from censorship to propaganda, but in general it tries to manipulate the perception of the receiver to benefit the original sender. Consider a situation similar to the uprising in the Egypt where rebels spread the news about an upcoming protest. To confuse the residents the government could spread information that the protest was cancelled or moved to multiple locations. The police can control the protesting crowd better as some residents are at home or at the wrong location.

The effect of disinformation is dependent on the number of receivers. To increase the audience it can be broadcasted in large volumes, or *spammed*,

to all members of the network. As the amount of spam increases the delivery delay of normal message also increases. *Spam suppression* is needed to keep the service usable by normal users.

As said before a computer is not capable of distinguishing between spam and normal messages based on content. Instead, information about the *sender* can be used to classify if a message is spam or not. The receiving device should *trust* the sender before it delivers or forwards a message. Trust can be determined by the user or based on interactions; cooperative devices can be trusted while new devices should earn their trust.

1.1 Problem Statement

- How can trust aid in the suppression of spam in a mobile social gossiping network?

Currently it is possible to create an ad-hoc communication network, but it has a well-known drawback. Spam suppression is needed to create a censorship-free social network environment. Ad-hoc networks should be able to withstand spam attacks by malicious users and provide a robust method of communication to trusted users.

This study focuses on achieving spam suppression in mobile opportunistic gossiping networks for a Twitter-like service. Twitter includes the ability to broadcast a message to all interested users. The framework can be applied to other networks; however, this is outside the scope of this work. Mobility and social relations differ from application to application and therefore comparisons between different versions are not included.

1.2 Methodology and Organisational Description

An implementation of an ad-hoc network is difficult, as it requires many devices and users to test all capabilities and features. Therefore to achieve the goal of creating a PSN that is able to withstand spam attacks a simulation is needed. The discrete time simulator NetLogo [55] is used for this purpose as it provides a multi-agent programmable modelling environment. It provides not only numbers and graphs, but also a graphical user interface depicting the simulation. The simulation gives an insight in the spam problems that ordinary PSNs have. It will also be used to validate the proposed solution.

This thesis is structured as follows: background information on PSNs and topics such as mobility and resilience against attacks are presented in Chapter 2. Followed by Chapter 3 in which we have written a detailed description and analysis of the proposed enhancement. Chapter 5 discusses

our simulations which we hope will provide us with insight into the effectiveness of Hearsay. In Chapter 6 we analyse the results of the simulations and answer the problem statement after which we look at future works.

Chapter 2

Background

This chapter summarises the existing scientific work and their relation to the different parts of the research question. Section 2.1 contains background information on Gossiping Networks and Pocket Switched Networks and the differences with the Internet. Section 2.2 shows research on possible attacks and how to avert them. Finally in Section 2.3, we discuss the work of other researchers on our main research question.

2.1 Gossiping Networks

Viral or word-of-mouth communication has been used for centuries to transfer messages from one place to another. In computer networks this ancient method is called *Gossiping*, or *Epidemic Routing*, and described by Vahdat and Becker in [54] as random pair-wise exchanges of messages among mobile hosts ensuring eventual message delivery. A gossiping network using mobile phones has been labelled a *Pocket Switched Network* (PSN) [8, 16, 23]. The term Pocket Switched comes from the fact that the system relies on users carrying devices in their pockets. As discussed in Chapter 1 two users within communication range exchange messages and as members move around the message spreads throughout the network. The delivery *delay* depends on the number of interactions between members, the number of messages and geographic size. In practice a message is delivered to a large group of users and not the entire network. This is due to limited amount of memory and communication time between users.

An advantage of PSNs compared to the Internet is that it does not rely on fixed addresses and uses local connections. When nodes with fixed addresses reconnect to the network all intermediate nodes need to update their routing table to adjust the information on node. PSNs do not have a fixed structure because they can not handle frequent changes. Although the Internet allows for fast connections across large distances, sending a message to someone in the same room via the Internet usually involves intermediate connections,

usually outside the room. A PSN can directly send a message if the user is within range. This reduces the amount of traffic and the need for a central authority handing out addresses.

A PSN always needs more intermediate devices to construct an end-to-end connection. In dense gossiping networks, where a node has multiple connections to nearby nodes, a fixed address network can be simulated by constructing an overlay network [3,41,47]. An overlay network is an abstract network on top of the actual physical devices. A device within such a network could have a connection to another device, while physically the devices are not connected. Such a network is able to provide a stable end-to-end connection even if the communication links between a few nodes fail.

A characteristic of a PSN is a low average density making overlay networks unsuitable for this thesis. Multiple hops are needed to spread a message throughout a PSN and a user must have frequent interactions with a diverse group of other users to compensate for the low density.

2.1.1 Mobility

Without overlay networks and fixed addresses PSNs rely on mobility for the dissemination of messages. To create a large-scale network mobile users are needed to transfer the message to regions with low density. Several studies [15, 24, 34] have shown that there are numerous encounters between users on a daily basis which makes PSNs a feasible alternative. The papers also show that it is likely for a user to frequently interact with the same group of users. Therefore, the physical interactions are not only a result of mobility but also because of the social relations between users. A social mobility model has been proposed [37] to simulate such an environment.

Heinemann [21] validated with real-world mobility traces that an opportunistic network is feasible. Such a network does not require user interaction to transfer messages between devices. The mobility traces consisted of users and the cell tower they were connected to. Once multiple users were connected to the same cell tower the simulator created a virtual world with the size similar to the radio range of the cell tower. Mobile users within this world exchanged messages whenever they were in range of each other. An initial message was sent by a single user and the results show that for Bluetooth communication (10 meter radio range) within one week 50% of the users had received this message. With Wi-Fi (100 meters radio range) 90% of the users had received it within 48 hours. As this simulation is based on traces of a hundred users delivery ratios can be higher in an environment with a different density.

2.2 Resilience

The problem statement in Section 1.1 focuses on suppressing spam in a Pocket Switched Network. However, spam is not the only attack that is possible. A PSN is based on user contributions and it has no central authority, giving an attacker multiple methods of disrupting the service. Resilience implies that when such an attack happens the effect will not spread throughout the network and the service will gradually return to a normal state.

Work by Jain and Sagar [26] and Kapadia et al. [28] lists many possible attacks on a *mobile ad-hoc network* (MANET), which is a broader term that covers all mobile networks with devices connections via wireless communication links. Possible attacks include:

Eavesdropping Listen to other messages and intercept conversations or passwords

Black/Grey Hole Drop all or a selection of received messages

Radio jamming Disrupt the wireless channel for all devices within range of the jamming device

Data tampering Adjust the contents of a message

Impersonation Send a message signed by another user

Denial of Service Cause disruption to the network by sending incorrect or large amounts of data

Not all attacks apply to, or are in scope of, this research. An example is eavesdropping as all messages are expected to be public in the Twitter-like network so no full encryption is needed. Black/Grey Hole attacks also do not affect the network as they can be considered as users who do not contribute to the network. Radio jamming hardware can be found on the Internet with ease[1], but as most available hardware can only jam signals in a hundred meter radius this can be considered as a minor annoyance to the users nearby. The remaining attacks have a serious impact on the network performance and are listed below.

2.2.1 Impersonation and Data Tampering

Public/private key-pairs provide a good solution against impersonation, data tampering and eavesdropping. However, verifying the sender in a centralised [17] or distributed manner cannot be used in a PSN due to mobility and the sporadic connections to neighbours.

[1] `www.jammer-store.com`

Fully distributed trust models for MANETs [12, 38, 46, 49] have requirements that are not feasible in PSNs, as they rely on neighbours or end-to-end communication. A solution assisted by the user, for example requesting both users to enter the same number, would ensure that the public key is transferred via a secure link. Without a secure link an attacker could intercept the key and replace it by its own. Such an act is known as a man-in-the-middle attack and results in the receiver trusting the attacker.

One such example of a user-assisted solution is a key signing party, which is a social event where people verify the identity of someones PGP key [17]. In 2012 a key-signing party was held at the Free and Open Source Developer Europe Meeting where 165 people exchanged their keys and passports to verify the credentials[2]. After the meeting each users uploaded the verified keys to one of the PGP keyservers. PGP uses central servers to distribute keys but a trusted key can be transferred to other trusted friends, ensuring a quicker distribution of public keys.

Since impersonation can be resolved by using user-assisted techniques this thesis focuses on providing a solution for denial of service attacks on a PSN.

2.2.2 Denial of Service

Denial of Service (DoS) is related to spam as it is one of the methods for disrupting a service in a manner that renders it unusable [5, 25, 56]. This is easily done on an unsecured PSN, an attacker just needs to send many more messages than normal people would. By the sheer volume of messages users will now predominantly see the messages sent by the attacker and they will spread these throughout the network. Gavidia [18] uses probabilistic verification to limit the spread, but this research is based on a static mesh network with public/private keys and not a mobile network.

A different approach is to focus on the sender of the messages. If the public key is included users can be identified and this information could help find the attackers. A DoS attack can originate from a small group of users from within the network. Other users can deny communication from the attackers once they have been marked as such. It is hard to identify the spam and label the user but coordinating users to block a spammer is even harder. If a user would report a spammer as such other users could accept this information and block it as well. However, a spammer could use the exact same method to block all other users.

Trust based solutions [6, 36, 48, 53] do not focus on finding the attacker but on the cooperating users. Each action of a user, such as forwarding messages, is judged by others and results in a reputation. Based on the reputation a user trusts another user and establishes a communication link. An attacker is required to gain trust before it can disrupt the network, but

[2] `https://ksp.fosdem.org/files/ksp-fosdem2012.txt`

once it does this will have a negative effect on its reputation reducing the impact of the attack. More details about trust and its uses can be found in [57].

2.3 Related Work

In this section we will focus on papers that are related to Hearsay. Papers are grouped by the two topics from the problem statement, Mobile Social Networks and Ranking and Trust. The first group focuses on projects that have created social PSNs. The latter group which discusses trust based solutions to a Denial of Service attack.

2.3.1 Mobile Social Networks

Shah [51] shows in his MSc thesis that a Twitter-like service based on a PSN is feasible. The main idea in this thesis is that users subscribe to their interests, *hashtags* in Twitter terms, and when they meet other users these profiles are exchanged. Based on a user profile the sender transmits some or all messages that could be of interest to the user. The benefit of the system is that the only required user interaction is the selection of interests; other activities such as transmitting profiles and messages are done in the background.

Messages with hashtags that are not of interest only get transferred if there is a high probability of contacting someone who might be interested in this tag. To support this every user logs the number of hashtag occurrences in profiles it sees. A notion of time is added to remove unwanted hashtags, as otherwise a user will remain interested in all hashtags.

Overall it gives a detailed report on how to create such a service, but as it is only is a proof-of-concept the implementation only works in a perfect world. The system has no defence against malicious users; so all attacks described in Section 2.2 apply. As profiles are essentially broadcasted to all listeners any attacker could reply with a spam message containing hashtags from all known profiles.

Research done by Mtibaa et al. [35] is more focused on showing correlation between interactions and the social graph. Message forwarding can be adjusted based on social information. Experiments were done during a conference with 28 participants where the device would vibrate if it detected a friend. It created a basic social network, but instead of transferring messages it could only alert users of nearby friends.

Hui et al. [23] discussed the challenges and feasibility of a PSN of which security was mentioned as an important area with many opportunities for innovative work. Haggle [50] was a continuation of this research, presenting a set of architectural principles for PSNs. However, it only mentions the need for security in the future work.

Another project that has built a social PSN is MobiClique [42], which uses an existing social network to bootstrap the network and uses the ad-hoc network when the user is not connected to the Internet. Using existing social networks can result in a quicker setup, but such a solution cannot be deployed in a situation without infrastructure. As this thesis tries to provide a communication tool in an unstable situation, like disasters or revolutions, any dependence on external systems should be avoided.

Overall these papers show that a Pocket Switched Social Network is possible, but it still has the same drawback as a normal PSN; no resilience against DoS attacks. The following subsection discusses multiple trust-based solutions and whether or not they can be applied to a social PSN.

2.3.2 Ranking and Trust

A trust-based solution is proposed by Samavati et al. [48] for the AMLeT framework. It introduces a hard and a soft trust model and individual nodes choose which model or combination they apply to each node they encounter. Using the soft model the system becomes optimistic about other nodes and gives them higher trust values. Hard trust is the reverse where a node should prove itself to be trustworthy and a single fault will have a strong impact on the trust value. The trust value calculated is based on interaction experience and creates a hardness factor for a specific node.

The trust value can be used to filter or limit the number of messages send to other nodes, or even reject communication with a node. Messages of trusted nodes will have a higher chance of being transmitted, while messages of possible attackers will not infect the network. Behavioural evidence is used to establish or revoke trust, however, evidence can be tampered to increase the rank of a user. The idea of AMLeT sounds reasonable, yet the paper is too short to provide answers such as how the network is bootstrapped or structured.

Another example is CONFIDANT [6] that uses trust and reputation to determine network routing in Mobile Ad-hoc Networks (MANETs). Trusted nodes are used to forward messages along a path to an intended receiver. Behaviour of nodes influences their trust levels and during an attack users notify others by sending out an alarm message. Based on the sender of the alarm message and other alarm messages the trust of the attacker is reduced.

PeopleRank [36] is a technique to determine the rank of nodes to reduce the number of communications. Nodes with a high rank are more interesting as they interact with more nodes. This increases the probability that a high ranked node will deliver a message. Although this system uses ranking it is not used to prevent attacks on the system. In the paper it is mentioned that nodes exchange 'their current PeopleRank values'; meaning that an attacker could report a high value to dominate the network. As the nodes forward any message without restriction a DoS attack is easily done, even for an

attacker with a lower rank.

A solution not directly related to the field of PSN is TrustRank [20], based on PageRank [40] which is used by Google[3] as its main search algorithm. The idea is simple, nodes are trusted if they have a relation with a node that was found to be trustworthy. Few nodes belong to the initial set of trusted nodes and each node gives an equal portion of its trust value to all nodes it has a relationship with. The result is that a node who is close to an initially trusted node will have a higher trust value than a node further away. And if a lot of nodes trust someone it will have a higher trust value. We will discuss this technique in more detail in Section 2.4.

Closely related to TrustRank is NodeRanking [44], but nodes only store a small proportion of the network and calculate rankings by communicating with their neighbours. As a PSN has no guarantee on communicating with a neighbour this solution is less suited for Hearsay. There are more propagation models for trust and reputation that can be found in [60] and [52].

Trifunovic et al. [53] have built a model for social trust in opportunistic networks. It uses both explicit and implicit trust to determine a trust value of a member. The research focuses on reducing the number of *sybils*, or identities, users can have. A sybil attack [13] is used by a user to adjust or abandon its reputation. The explicit trust is based on relationships declared by users and a value is assigned based on their connectivity in the social graph. Direct friends are assigned a score of one, which they distribute among their friends who are not friends of the user. The implicit trust value is based on similarity and familiarity, where a node its interactions and friendships are compared to other nodes. As stated in their future work there is no research done on how to weight explicit and implicit trust. Another problem is that nodes rely on values calculated by other members, this dependency can be exploited to gain trust. The method of calculating explicit trust assigns the same score to all direct friends. The trust value is also distributed to nodes one social hop distance further away from the user, while relations between users on the same hop distance are ignored.

In this section we have discussed different methods to achieve resilience against attacks and our thesis focuses on spam suppression, as stated in the problem statement (Section 1.1). The AMLeT research, Trifunovic, and TrustRank are the most interesting research papers for achieving this goal and a combination of these approaches would provide a method of limiting outgoing messages based on trust relationships.

2.4 TrustRank Explained

The AMLeT framework [48] proposed limiting messages based on a value. The value is calculated based on interaction experience of nodes, also called

[3] `http://www.google.com`

implicit trust. The research of Trifunovic et al. [53] uses both explicit and implicit trust to determine a trust value, but their method of calculating the *explicit trust* does not use all information in the social graph. *PageRank* can be used as a replacement as it has no specific requirements on the graph.

PageRank is a method of calculating a value based on a measure of interest. The interest of a page is calculated by counting the number of other pages that have a link to this page. It is no more than logical that if a page has more incoming links in comparison to other pages its content must be more important. A link represents a value, but counting the incoming links is not the same as its rank. If a page with a high rank has a link to a page without any other links, it is likely to be more interesting that any other page with a single link. Rank gets transferred between pages according to the number of links that they have.

In short the PageRank algorithm awards each web page a start value. In each round a part of the value is distributed according to the number of outgoing links on that page. If many of pages point to the same page the value of this page increases as it gathers more value.

PageRank has been altered and extended by many researchers but one version is of particular interest to our research, *TrustRank* [20]. TrustRank is a biased version of PageRank, which means that instead of giving every page a single start value only the trusted pages are given this value. By default pages will get assigned a trust value of zero which can only be increased if other pages who do have a value link to this page.

2.4.1 Computation

TrustRank and other PageRank algorithms use equation 2.1 on a directed graph $G = (V, E)$ and iterate it M times to reach convergence. The number of iterations depends on the size of the graph and the authors of the original PageRank paper *PageRank* used 52 iterations on a graph of 322 million links and 24 million pages.

$$\bar{r}_{i+1} = \alpha T \bar{r}_i + (1 - \alpha)\bar{d}, with\ 0 \leq \alpha \leq 1 \tag{2.1}$$

$$d_i = \begin{cases} 1 & \text{if node } i \text{ is trusted} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

In equation 2.1 α is the decay factor, or the percentage of trust a node propagates. T is the *transition matrix* and d the *bias*, also called the initial trust. The value of r_0 equals d, which is created using equation 2.2. All trusted nodes get a starting trust value of one, while the rest receives zero trust.

To create the transition matrix T for every possible link $p - q$, with $p, q \in V$, an entry is added:

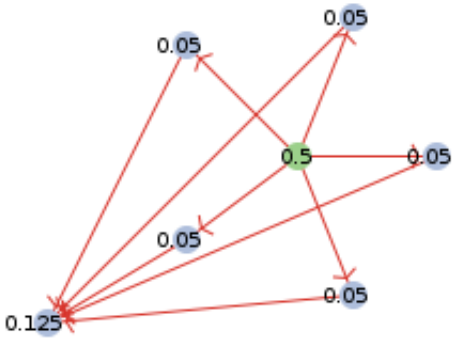


Figure 2.1: A network with 5 friends and one common node, labels are the TrustRank scores with $\alpha = 0.5$. The central node is trusted and distributes it trust value to the surrounding nodes, while the common nodes receives trust from all friends of the trusted node.

$$T(p,q) = \begin{cases} 0 & \text{if } (q,p) \notin E \\ 1/\omega(q) & \text{if } (q,p) \in E \end{cases} \tag{2.3}$$

In this equation $\omega(q)$ is the number of out-links node q has. If q has a trust relationship with, or links, four nodes the number of out-links equals four. The value represents the amount of trust is transferred from q to p.

$$1 - \alpha \leq \sum_{i=0}^{|V|} r_i \leq \sum_{j=0}^{|V|} d_j \tag{2.4}$$

In equation 2.4 r_i is the trust value of a single page. The total trust of the graph is less or equal to the sum of the initial trust, but equal or higher then $1 - \alpha$. As nodes with no outgoing links still give a part of their trust (α) away a graph of a single page with no outgoing links has the lowest sum of $1 - \alpha$. A graph where every page has an outgoing link will not lose any trust and will have a sum equal to the sum of the initial values of all pages.

2.4.2 Example

Figure 2.1 is used as an example and the final TrustRank scores is visible on each node. The network consist of 7 nodes and the node in the centre is the only node who is trusted. From the network and the set of trusted

nodes we can determine the following values:

$$d = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The value of d means that the first node (the green central node) is trusted while the rest is not. The bias can be adjusted to give another node more trust. In the figure the final score with $\alpha = 0.5$ and $M \geq 3$ is visible on each node. The calculation starts with the green node having all the trust while the others have nothing. In the following round it gives α times its trust away and divides this over all out-links. Now the central node has 0.5 and the five trusted nodes have 0.1. During the second round the leftmost node receives trust from the five neighbours, $0.1 \times \alpha \times 5 = 0.25$. Finally the leftmost node gives half it trust away, but as that node has no out-links this is lost.

Hearsay relies on TrustRank for computing the rank of users. The rank of a user is used to control the order in which messages are propagated. Sending spam throughout the network requires an attacker to adjust the social graph in such a manner that it is trusted by a large number of users. With the rank computed via the outgoing edges of users an attack can only happen if there are user who trust the attacker.

Chapter 3

Design

Hearsay is a combination two techniques, a PSN using TrustRank to suppress spam. Gossip networks are by design vulnerable to attackers sending spam. The solution proposed by AMLeT [48] limits the outgoing messages, combined with the explicit trust of Trifunovic et al [53]. is the basis of Hearsay. The goal of our thesis is to achieve spam suppression by promoting messages of trusted users.

3.1 Design Considerations

Hearsay assumes the same conditions as a regular Pocket Switched Network. A device is carried by a user with a radio capable of establishing opportunistic connection with nearby devices. An opportunistic connection does not require the user to accept the transmission or enter security credentials. Further considerations to achieve spam suppression in a mobile social gossiping network are:

- Nodes should not rely on global values or values calculated by other nodes, such as a global average, as these are subject to attacks.
- New nodes should be able to receive data to update the local social graph, or *local view*, but sending data should be limited in order to prevent sybil attacks.
- The number of spammers is unrestricted, so that Hearsay can be used by small rebellious groups.
- Ranking techniques that focus on finding associates of known spammers are outside the scope of this thesis due to the size of the known spammer set.
- Network types, other than a PSN, are not considered.

3.2 Hearsay

As stated in Section 2.2.2 spam could be used to create a denial-of-service attack. Spam can overwhelm normal messages leaving users no other option than to leave the network. To overcome this well-known challenge in PSNs spam messages should be assigned a lower priority than normal messages.

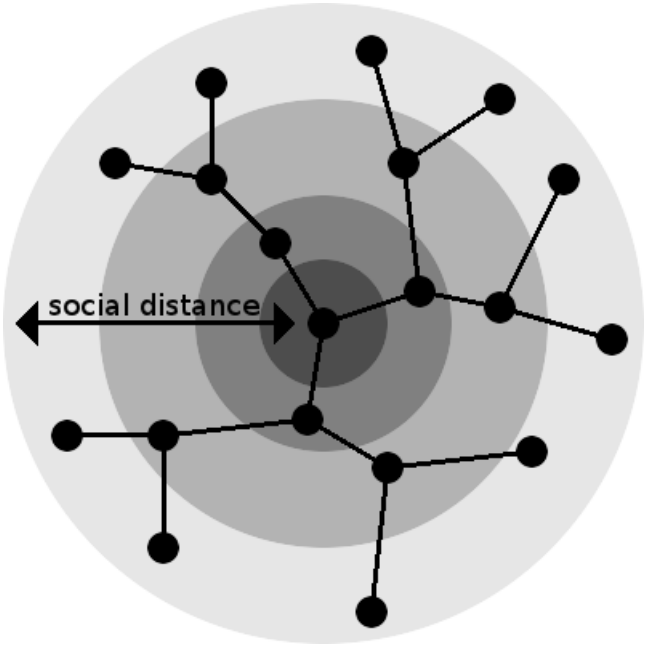
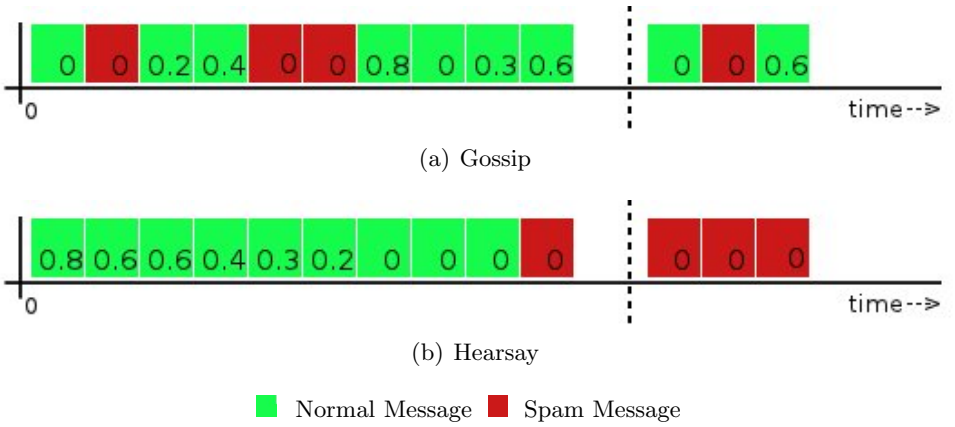


Figure 3.1: The social distance on which the ranking is based. A greater distance results in a lower ranking.

From the perspective of a user messages have a certain value; a message from a close friend is valued more than a spam message. Hearsay is built upon the idea of promoting messages sent by users close in the social graph. Figure 3.1 depicts the social graph and the related social distance. Friends, or trusted users, are users within a short social distance from the user, while attackers will have a larger distance. Friends can help by forwarding a message of a trusted user to other nodes. This will improve the spreading of the message to other trusted users. With limited communication time a user needs to decide which friend it will help first before the connection breaks. A ranking of friends decides the order in which messages will be send. This ordering indirectly demotes spam messages, as spammers are often not friends.

Figure 3.2 shows a communication for both gossiping and Hearsay. Each block is a message with their perceived value, or *rank*. Gossiping has no knowledge of the rank and sends the message according to the order it received them. While Hearsay sorts the outgoing messages and starts by send-



(a) Gossip

(b) Hearsay

Normal Message Spam Message

Figure 3.2: Messages sent during communication with their perceived values. Hearsay prioritises high valued messages, reducing the amount of spam sent before the transmission ends after ten messages.

ing the highest ranked message. If the communication ends after the tenth message gossip would have sent three spam messages, while Hearsay only sent one.

3.2.1 Ranking Friends

The rank is based upon social relation between users. A trust relation is directed and personal [57], meaning that each user has a different view of the relation and has a different ranking of users. The term *social graph* will be used for the social relations between all users, while *local view* will be used for the relations known to a single user. These social relations are explicitly defined by the users, for more information see Trifunovic et al [53]. Each user declares who it trusts and broadcasts this to all other nodes. Upon receiving friendship information the devices stores this information in its local view so it can be used to calculate the ranking. The social graph is a representation of users, the nodes, in the network and their relations, the edges, see Figure 3.1.

To calculate the ranking based on this idea, TrustRank ([20] or Section 2.4) is applied to the local view. As a starting point of the algorithm the only trusted node in the graph is the user itself. In each iteration a part of the rank each node of a user has is divided among its neighbours. Close friends of the user will be highly ranked while others will be ranked lower. Each relation in the graph represents a recommendation and highly recommended users will have a higher ranking even if the distance in the graph is similar. More information on how to calculate the rank can be found in Section 2.4.

Trust is directional and therefore it does not help to have many outgoing relationships. One must have friends who declare their trust and with more

recommendations others will assign a higher ranking and are more likely to forward the message.

With TrustRank applied to the local view Hearsay is able to rank messages according to the social relations between users. This ranking can then be applied to order the message queue and promote messages of friends.

3.2.2 Message Propagation

Messages of friends have a higher priority during communication, and a paper by Miklas et al. [34] showed that two thirds of the daily encounters are between friends. Due to an overlap in friends a user is therefore likely to receive interesting messages.

Users with many friends will be able to communicate their message to more users as more friends forward it. On the other hand a user with a single friend will have difficulty sending a message as all other users are unlikely to forward it. A message therefore only spreads if the friends, or friends-of-friends, of the sender have given it a high rank or there is enough bandwidth available. This and other propagation speed properties are studied in Section 5.3.

As PSNs are based on ad-hoc communication a message, in a realistic scenario, is restricted to the geographical area surrounding the sender and its friends. Current social networks provide a global communication platform, while with Hearsay it is only possible to communicate to users who are within the social neighbourhood. As the social distance increases the probability that a message will be forwarded decreases. Spam will for that reason remain within a small community. Once users start revoking their trust relation with the spammer the average rank of the spammer will decrease, reducing the size of the effected community.

Compared to gossiping, Hearsay will spread information slower as forwarding is limited by the social graph. The overhead in transmitting friendship relations will also effect the propagation speed of normal messages. However, the number of normal messages will be far greater than the social graph update messages. Due to the differences between the actual social graph and each user's local view propagation speed will be less as not every user has the latest information.

3.2.3 Spam Suppression

With Hearsay spam suppression is implied, as users who send spam usually do not get many friends. Therefor a spammer should convince other users to add him as a friend before sending spam. Once the spammer has gathered enough friends, it will reveal itself by sending spam. Users that have a trust relation with the spammer will contribute by forwarding the messages. They will also have trouble viewing messages of normal users as the spam

is overwhelming them. Blocking the sender will remove all received spam messages from the device and revoke the relation with the spammer. Other users will receive the update and alter their rankings, rapidly demoting the spammer. This behaviour and other spamming properties are studied in Section 5.4.

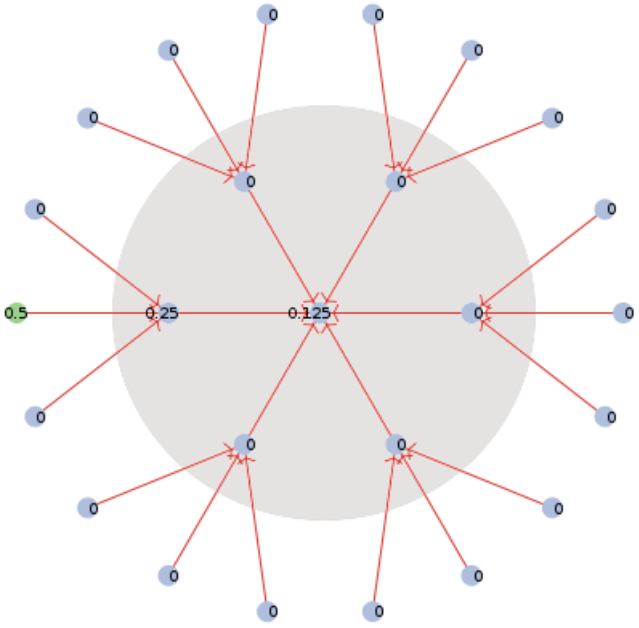


Figure 3.3: A group of attackers in the centre, with normal users on the outer ring and the labels note the ranking. Only relations directed towards the attacker are drawn. The user has no direct relation to the spammer, but the spammer receives some value.

Figure 3.3 illustrates an example of a group of spammers working together. The ranking is calculated for the leftmost user, although other friends of this user are not depicted. The group forms a social circle around the main active spammer to increase its rank. Spam will initially be forwarded by members of the group to the outer ring of users. Group members are therefore passively supporting the spammer. If there is enough bandwidth available unsuspecting users will forward the spam to an even larger group of users. Users must now reevaluate their relationship with the passive member of the spam group and revoke it even if it has not send spam itself. With a single passive group member blocked the rank the active spammer receives via other group members has not decreased, creating an opportunity to send more span. To block the spammer each user with a relation to a passive group member needs to revoke it, otherwise the attack group just replaces the spammer with a new sybil.

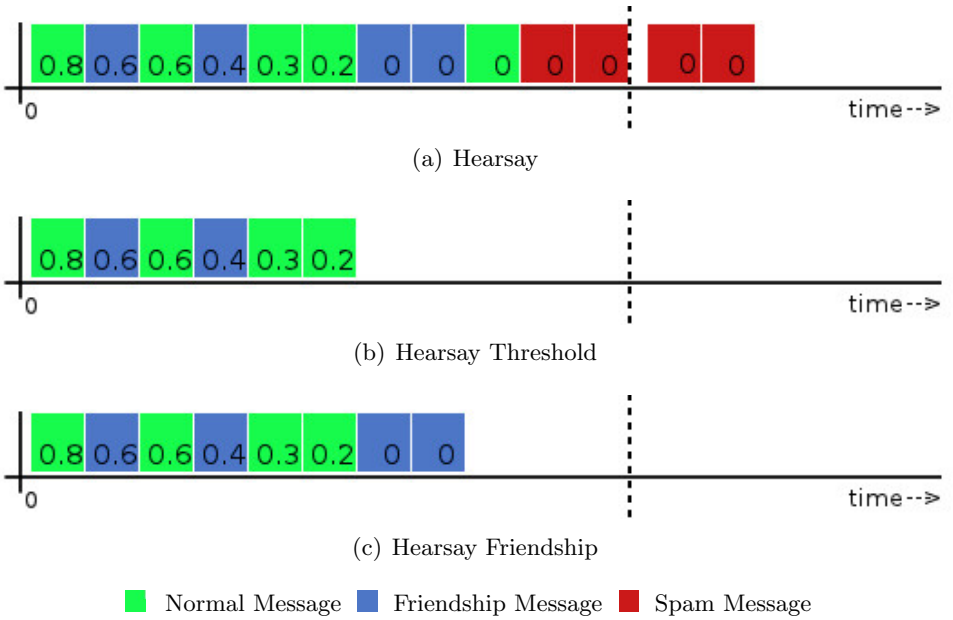


Figure 3.4: Messages sent during communication, restricted versions. Figure 3.2 contains the original version containing gossiping and Hearsay. Hearsay Threshold removes all messages with a rank of zero, while Hearsay Friendship looks at the type of a message. If the message contains social graph updates it is still sent in the remaining period before the connection breaks

With the information stored in the local view it is possible to find members of the spam group. A simple message such as "You received this message via Alice" or "Charlie, friends of Alice, says ..." could inform the user who might be helping the spammers. Other techniques, such as Anti-TrustRank [31], that reduce the ranking of spammers could be implemented to reduce the spam via multiple social relatives, but these techniques are outside the scope of this research.

As the spammer is not directly related to the users it is attacking the rank will be lower than close friends and many friends of friends. This reduces the effectiveness of such an attack as lower ranked messages are less likely to be forwarded. Also each member of the attack group needs multiple relations with other users which takes time, while each user can quickly block an attacker. This type of attack is therefore not very effective in disrupting the network.

3.2.4 Limited sending

Hearsay sorts all messages to maximise the total value of the outgoing messages. Another strategy, shown in Figure 3.4(b), is to limit the list of mes-

sages to only known users. A message from someone outside of the local view can be dropped as it is not likely to be interesting. To decide if a message can be dropped one can examine the TrustRank score, if this is equal to zero the user is either outside of the local view or has no trust relation with the trusted user. We call this strategy *Hearsay Threshold*. The value of the threshold can be higher than zero, which will block users with a low trust value, although this is not considered.

Forwarding messages is at the heart of a PSN, limiting it would slow the network down. To improve the propagation of messages the remaining communication time can be filled with friendship messages of untrusted nodes. In Figure 3.4(c) the improvement is visible as it now includes more messages that contain friendship information. Adding these messages keeps the local view updated on new users and relations, without sending possible spam messages. To make a clear distinction between the previous versions this strategy is named *Hearsay Friendship*.

3.2.5 User Identity

To assign rankings to users each user must have a unique identity. This identity is used for creating relations as well as sending messages. As discussed in Section 2.2.1 public/private key pairs are a solution to prevent impersonation. The public key is the user identifier while the private key is used to sign messages, which makes it possible to verify the sender of a message.

Social relations between users are declared by each user independently. Both create a message containing the public key of the trustee and its own and sign it with their private key. When this message is received by any user it has enough information to safely insert the relation into its local view.

Hearsay does not require a centralised component as there is no need to check the credentials against an actual person. The public key is used only to identify the source of the message and store relationship information in the social graph. Other information such as name or phone number can be exchanged inside messages, during a meeting or via a centralised component, but this is not needed for Hearsay to work.

Although the system does not require user information it does exchange public keys and social relations, which can be used to identify a user and harm its anonymity and privacy. The system is built on stable relations between users, identifying a single node could result in the identification of its friends. To link a public key to a user an attacker could follow a user until it is not surrounded by other users. The attacker will now be able to associate the public key with the real identity of a user.

Another method of revealing information about the users is a neighbourhood attack, discussed in [59]. If one knows a partial social structure it can be mapped to the social graph. An example is when the social graph

consists of ten users and the attacker knows that Alice has five friends in the network, then he could limit the number of possible public keys to those of members with five friends. Using more data sources could result in a complete mapping between the real world and the virtual identities.

Therefore, Hearsay provides a messaging service without guarantees on anonymity. The focus lies on spam suppression via social relations for which an identity is needed.

3.2.6 Bootstrapping

For a node to enter the network it can ask its friends to trust him, which creates a link from the friend to the new user. As this declaration of trust is communicated throughout the network other nodes add an entry in their local view and assign a rank to the new user. In a Pocket Switched Network messages could take a few hours to be delivered to a large section of the network. In this time messages sent by the new user will likely not be forwarded as its score is very low. However, since friendship message of itself are sent before messages of other users the local view is likely to be updated at the same time as the message is received. Only users with no direct contact to the new user will probably not forward the message.

Implementing a social network using Hearsay would be an ideal match, but this method can also be applied to a system without user interaction. Friends can be selected opportunistically based on experience, for example a node who interacts on a regular basis could be considered a friend. The result would look similar to the research done by AMLeT [48], although it only works with local experiences to determine value.

3.2.7 Resources

Mobile devices are limited in resources, energy being the most important. Storage has been limited, but a recent trend is to increase the storage to multiple gigabytes without requiring external media. The expected storage of a very large local view is still less than one gigabyte. As an example one could store one million users, with each user having 200 friends and a 32-bit integer as identification, in a database of 0.748 gigabyte ($(1,000,000 \times (200 + 1) \times 32)/2^{33}$). A graph of this size would be largely unused as the user is only interested in friends within a few hops and with a high score.

Hearsay uses TrustRank which requires multiple matrix multiplications causing a high load on the processor. Limiting the recalculations reduces this load and it is likely that the nearby part of the social graph doesn't change too often. As TrustRank is based on the well studied PageRank algorithm there are many techniques to limit the CPU load, like local updates and Monte Carlo estimations [4].

Trust is only calculated from a local perspective instead of determining

the ranking on the receiver. This would require a complete recalculation of the entire graph and the benefit would only be marginal due to the high number of encounters with friends. When meeting strangers it would help, as high valued messages are not from its social neighbourhood. But this could be exploited by attackers by triggering recalculations and wasting a users battery.

Any message it includes the actual data, a public key of the sender and the hash calculated by the private key. All this information is used to determine the sender and verify if no one has tampered with the data. The overhead of signing and including the public key is 2304 bit if one is using 256 bit for the SHA-2 hash and 2048 bit for a public RSA key. These values are common for signing messages as it is unlikely that they can be broken within the next few years [43]. If the message would be a Twitter message of 140 ASCII characters the overhead would be 288 characters. Some optimisations or different choice of algorithms could provide a lower overhead but that is not within the scope of this research. A simple solution would be for a sender to label the public key and use the label instead of the key. A receiver could then recreate the original message linking it to the key it received earlier. A similar technique could be used to reduce storage space on the device. For limited devices a notion of time could be included to remove old messages and save disk space, but this would require some knowledge on how fast messages propagate.

Although Hearsay uses a lot of resources on the device developers have many options as discussed above to reduce its footprint by selecting different techniques and optimisations.

Chapter 4

Implementation

Simulations are needed to test the feasibility and effectiveness of Hearsay. An actual implementation on mobile devices is too time consuming as it requires many users and a complex social network. Instead, we explored the properties of Hearsay with the NetLogo simulator. NetLogo [55] is a visual simulation tool for multi-agent environments. It features a simple language that allows a programmer to easily define nodes and their interactions. Included is an interface builder to show the virtual world during simulation and adjust simulation parameters.

NetLogo was chosen because of previous personal experience in the field of modelling gossiping algorithms. Visual feedback provides insight into the model while it is running a specific test. Multi-agent environments often exhibit emergent behaviour where a rare situation triggers a corner case. Visual feedback helps to analyse such events by illustrating the current state of the network and the nodes that have caused the problem.

In this chapter we will state, in Section 4.1, an overview of all simulation details, such as the size of the simulated area and the number of simulation runs. In Section 4.2 details are given on encountered issues with NetLogo. Finally, in Section 4.3 the credibility of the outcome is discussed.

4.1 Simulation Details

Each test in the next chapter represents a run of at least five times and the result is the average of all runs. For each of these runs the random seed of the test is altered, which results in different starting points, walking patterns and social graph.

Different tests in a single graph use the same set of random seeds. This ensures that each test is performed in the same conditions as the rest. By design NetLogo runs are reproducible as its pseudo-random number generator and agent scheduling algorithms are deterministic. However, NetLogo cautions that results may depend on the system configuration, which is

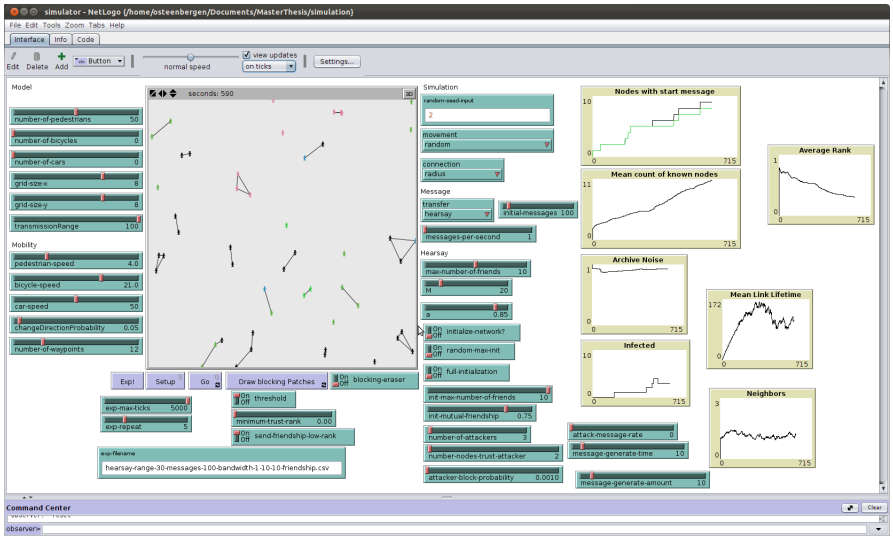


Figure 4.1: The NetLogo Simulation Tool showing the visual interface with the simulated area, parameter settings and graphs of the current test.

Simulator	NetLogo
Simulator Version	5.0-RC7
Java version	Java(TM) SE (build 1.6.0_32-b05)
Operating System	Ubuntu 11.10

Table 4.1: System Configuration

stated in Table 4.1.

A feature of NetLogo is the ability to publish the simulation on the Internet. The Hearsay simulation can be found at `http://onno.steenbe.nl/thesis` and shares the same code with the simulations done in Chapter 5. Some functionality has been removed, for example the ability to run multiple tests, but only features that do not effect the outcome of the simulation.

4.2 Implementation Details

The simulation details discussed above provide an insight into the inner workings of the simulator, but the implementation of certain parts have a large influence on the result. In this section we discuss some implementation details that are of influence, like mobility and the communication layer.

4.2.1 Mobility

The selected mobility pattern for testing Hearsay is Random Direction without pauses, see Camp et al. [7]. A node moves at a fixed speed in a random dir-

ection. After a random amount of time it changes the direction. In the paper by Camp this is described as "[the node] chooses a random direction and selects a destination anywhere along that direction of travel".

For testing purposes a separation between the mobility layer and the communication layer was created. This gave the opportunity to test different communication paradigms on the same mobility pattern. Reproducible tests require that the random number generator produces the same list of random numbers if the same *seed* is given. Every time a random number is needed the top number is removed from the list and used in the simulation. Without a separation between the mobility layer and the communication layer both use the same list of random numbers. Changing a parameter influences the required random numbers, which effects both layers. By using a different seed for each layer a parameter change will only effect a single layer.

4.2.2 Communication

Communication can be simulated in NetLogo by creating links between nodes. Each time slot every node checks for new neighbours within range and breaks links with nodes that have drifted away. A link can be directed or undirected and the latter was selected due to the nature of a PSN.

All nodes have two message lists. One for storing all available messages, the *in-box*, and one is used as an *out-box*. The out-box contains messages that the device wants to forward. Once a message has been sent to the current neighbourhood it is removed from the out-box. If the neighbourhood changed, the out-box content is replaced by the in-box content, causing the nodes to send their highest priority messages again. This guarantees that all nodes receive the highest ranked message. Without replacing the out-box contents new users within range are more likely to receive spam.

4.2.3 NetLogo Development

To implement the separation between the mobility and communication layers a modular system is required. NetLogo comes with a large library of simulations, but a modular system and Pocket Switched Networks are not included. Some techniques can be borrowed from the examples, such as creating nodes and communication links, but the NetLogo examples do not include large simulations with multiple interchangeable modules.

Figure 4.2 details a sequence diagram shows how a modular system was implemented in NetLogo. The main program includes listeners for basic actions, such as setup and send/receive messages, and enables modules to register their functions at the corresponding listener. To run a function the main program just needs to call the corresponding listener. Multiple functions can register to the same listener, which is useful in a situation like creating a new node or sharing code between modules. Sending a message

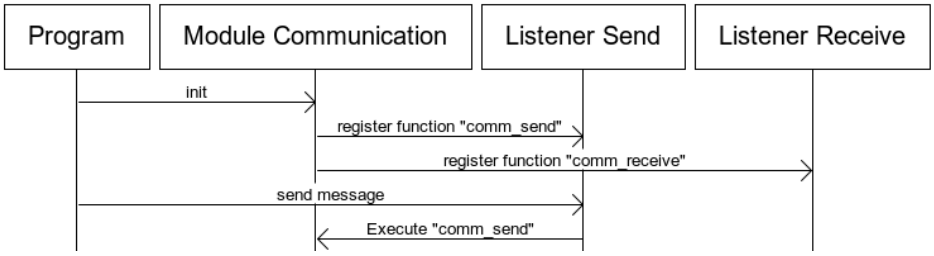


Figure 4.2: A sequence diagram showing how modules were implemented in NetLogo. The program includes listeners for each action that is implemented in multiple modules.

involves a generic gossip module that iterates over the outbox and informs the specific Hearsay module to forward a message. The Hearsay module is also responsible for sorting the out-box after a new message is received.

Hearsay simulations require a lot of processing power to calculate TrustRank values and simulate interactions. To reduce the number of calculations the TrustRank is only computed when a node has new social network information. To simulate a network with an existing social graph all relation are added at the start of the simulation. Each node then runs the TrustRank calculation on its local view to determine ranks.

To save computations one could store the calculated social graph and reuse it. NetLogo has a built-in option to export the current state of the simulation, but there is a known bug with the table and matrix extensions. Our simulation relies heavily on these extensions to calculate the TrustRanks and to keep track of assigned ranks. After discussions with a NetLogo contributor we decided that it would take too much time to implement a solution compared to the time saved during setup.

During development other bugs within NetLogo were found and reported to the maintainers of the code. Most bugs, such as incorrect handling of errors and underlying Java problems, were fixed within a few days and are included in newer releases of NetLogo. Development was done with a release candidate version instead of the stable version. However, all bugs encountered also existed in the latest stable version. The release candidate was chosen because of speed improvement in comparison with the stable release.

4.3 Credibility of the result

Multiple papers discuss the credibility of Mobile Ad-Hoc Networks (MANET) simulations [2, 29, 32, 33]. PSN is closely related to MANET, therefore the same credibility issues apply. Due to the complexity of an actual implementation all tests are done using simulations. To build a simulation some

aspects of the real world are simplified. An example used in our implementation is that the radio is modelled as a perfect circle around the user. Kotz et al. [30] show that in some cases this simplified radio model effects the outcome of the simulations. A more realistic model would decrease the number of interactions as the signal is distorted.

Repeatability is one of the issues most MANET papers do not address. In Section 4.1 and Chapter 5 all information needed to reproduce a test is available and the simulator is publicly available on the Internet.

Another issue that can not easily be solved is the lack of a confidence interval. Each test is done multiple times and the mean result is plotted in graphs. The confidence interval, or the sample standard deviation, illustrates how reliable the mean is. However, PSN simulations rely on mobility which effects the standard deviation and this effect is discussed below.

As stated in Section 4.1, each result is the average over five test runs. Each test run has different initial positions for the users and uses different random seeds, resulting in different interactions between users. Two user could have frequent interactions in the first run while there is no interaction in subsequent test runs. Therefore, the test outcomes differ greatly which results in a high standard deviation.

Figure 4.3 shows that even if the test is done 500 times the average standard deviation is very high. The test was an one-to-all broadcast using normal gossiping and the standard deviation was computed for the coverage of the start message. The standard deviation eventually drops as more tests report full coverage. With more test-runs the line becomes smoother as a single test has less influence on the result. The high standard deviation of 25% results in a confidence interval of 68% that the actual value is within one standard deviation below or above the computed mean. Although one could report with confidence that eventually full coverage is achieved, a statement about the first thousand seconds is less believable. This section of the test is most important as it shows how quickly a message spreads and how difficult it was to reach the last user without a message.

However, each test in one plot uses the same set of random seeds. This ensures that, even with a high standard deviation, the only difference between the test is the selected parameter. All nodes are initialised in the same location and have the same interactions. Therefore, the small differences between results are because of differences in communication, not due to mobility.

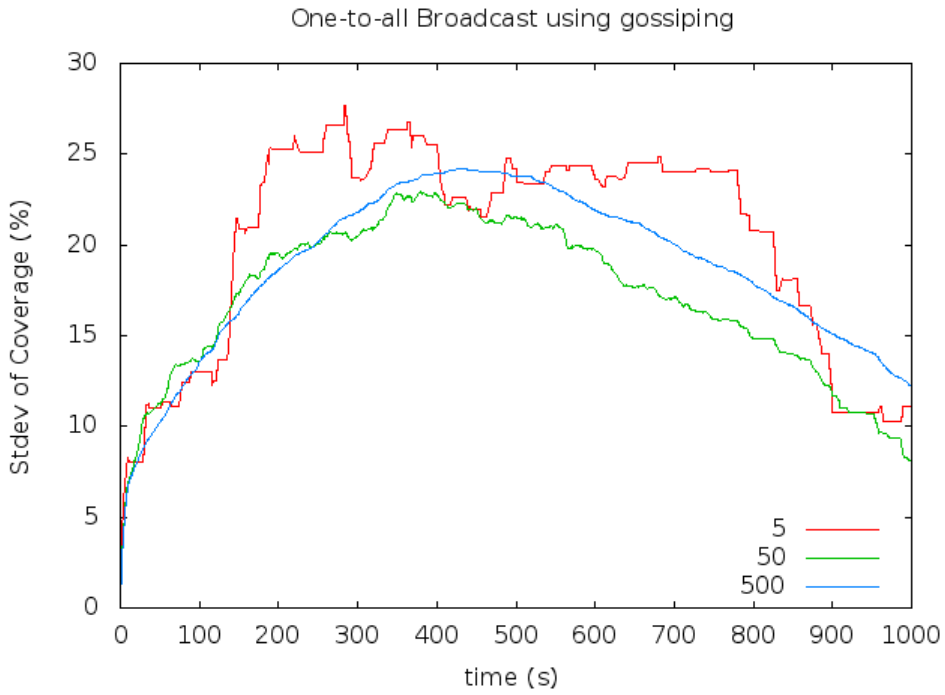


Figure 4.3: Standard deviation for coverage of a one-to-all broadcast using gossiping. Different numbers of test runs

Chapter 5

Evaluation

In this chapter Hearsay is evaluated using NetLogo [55] as described in Chapter 4. Section 5.1 discusses gossiping properties of Hearsay, like eventual delivery, and validates them against regular gossiping. Followed, in Section 5.2, by simulations that show that the system reaches a stable state. Message propagation speed using different parameters is discussed in Section 5.3. And finally in Section 5.4 the effect of hearsay on reducing spam is discussed.

A list of conditions is stated in Table 5.1 and is used for every test unless stated otherwise.

5.1 Gossiping

To build a social network it is of importance that messages eventually arrive. This section focuses on showing that Hearsay can be used to transfer messages.

First he influence of radio range is illustrated in Figure 5.1. A single message was propagated throughout the network using different radio ranges and the coverage is plotted over time. Communication time critically depends on the amount of time users are within range of each other. Wi-Fi, with a range around 100 meters, clearly outperforms Bluetooth (10 meter range). This is expected behaviour as with a shorter range the probability of a nearby user drops. For the simulations we selected the 100 meter radio range as Wi-Fi chips are common in mobile devices.

In Figure 5.2 the difference between gossiping, random walk and Hearsay is illustrated. For this simulation a simple one-to-all broadcast is used. A single user is selected at random to send the initial message. The vertical axis denotes the percentage of users that received this message.

Gossiping forwards a message to all neighbours and it is the quickest method to deliver a message to the entire network. Random walk is when a user only transmits the message once, after this the message is archived and

Number of nodes	50
Size of simulation area	$1km^2$
Shape of the world	Torus
Duration	$5,000$ seconds
Simulation runs	5
Mobility Model	Random Direction
Speed of the nodes	4 km/h (1.1 m/s)
Direction change probability	0.05
Pause time after direction change	0
TrustRank iterations (M)	20
TrustRank decay factor (α)	0.85
Maximum number of friends	10
Radio Range	$100m$
Radio Model	Circular Disc

Table 5.1: Simulation parameters

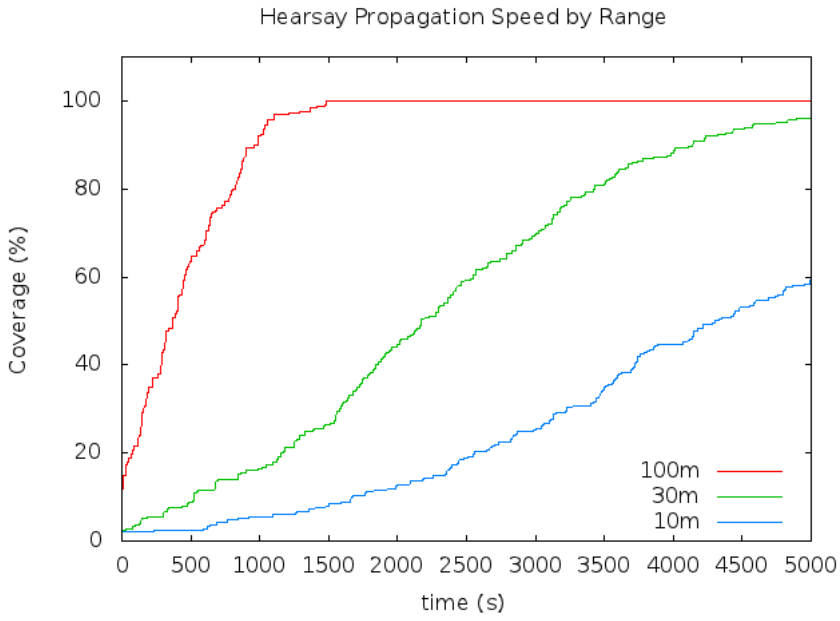


Figure 5.1: Propagation is slower when devices communicate at shorter radio ranges.

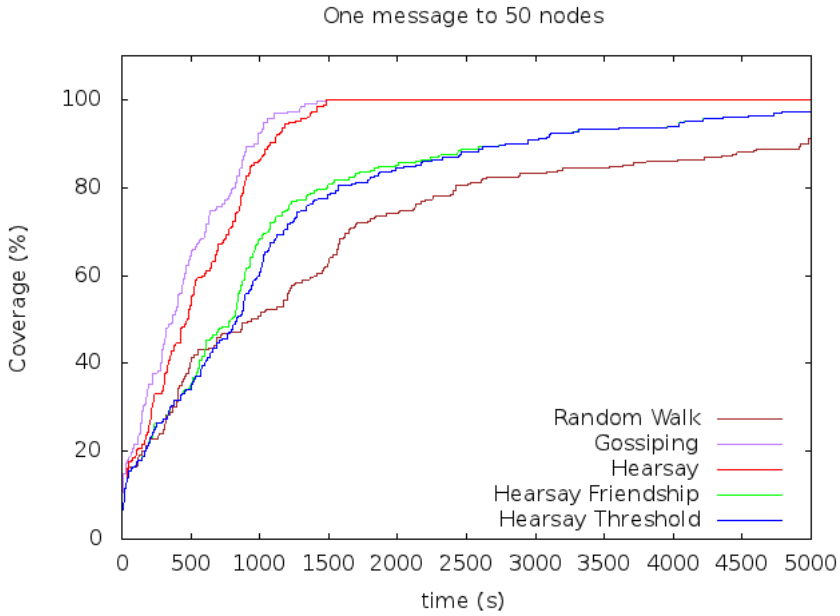


Figure 5.2: Comparison of protocols using One-to-All Broadcast

not retransmitted. This is the slowest method of communication because in the worst case there is only a single user in the network able to forward the message. However, as the message is broadcasted to all neighbours it can happen that a few users forward the message.

Hearsay transmits messages according to their rank, higher ranked messages are sent out first followed by lower ranking messages. Figure 5.2 shows that Hearsay performs just below standard gossiping. As the test only included a single message the delivery of messages will be slower if there are more messages of a higher rank. The difference with gossiping is due to the overhead in sending friendship messages.

Hearsay Friendship is close to Hearsay Threshold but below Random Walk during the start of the experiment. When the social graph is updated the performance improves for both versions. Hearsay Threshold limits sending of messages, even friendship messages, and as predicted in Section 3.2.4 this harms performance. Low ranked or unknown nodes are blocked and new relations are only received via existing relations or direct contact. Hearsay Friendship is only a marginal improvement over Hearsay Threshold, a larger difference was expected as Friendship includes friendship messages of untrusted nodes. However, based on this result it is likely that the time remaining to send the friendship messages was too short.

As noted before the simulation is on a network with one broadcast message and multiple friendship messages. In Figure 5.3 the results are shown of a simulation where initially 100 nodes, with 30 meter radio range, have 100

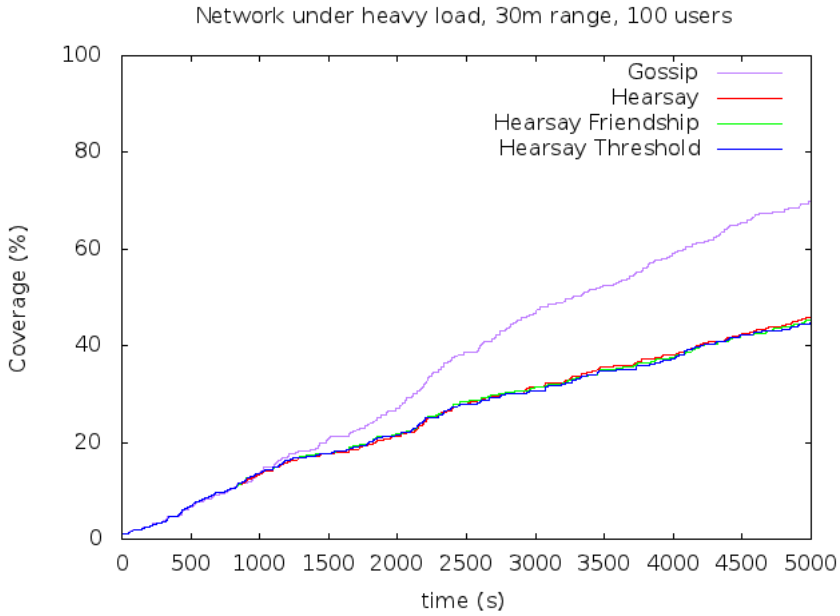


Figure 5.3: The effect of low bandwidth on propagation speeds. Different Hearsay versions perform similar due to the abundance of messages with a high rank. Gossiping performs better as it sends the oldest messages in the network first.

messages and every 10 seconds 10 new messages get added. The coverage of the first message is plotted in the figure. To simulate restricted bandwidth every user has a transmission rate of one message per second. Each communication link is fully used which results in a similar performance for all Hearsay versions. Even with a threshold there are more messages to send than what fits in the short communication time of a 30 meter link. The average time a link exists between two users is close to 25 seconds, which is enough to transfer a large number of messages using the latest technologies but to simulate restricted bandwidth the users only send one message a second.

Gossiping performs better as it uses a 'First-in, First-out' approach. The broadcast was one of the first messages in the system, thus the first to be forwarded to other nodes as users do not change the ordering of their in-box. Hearsay ranks each message so newer messages of higher ranking users get sent first resulting in slower propagation in comparison to gossiping. The size of the gap between gossiping and Hearsay is due to the number of new messages. During the simulation 5,000 messages are added to random users, resulting in an average of 100 messages per user. As predicted in Section 3.2.2 Hearsay performs slower than gossiping due to the social graph ordering and the overhead of sending friendship messages.

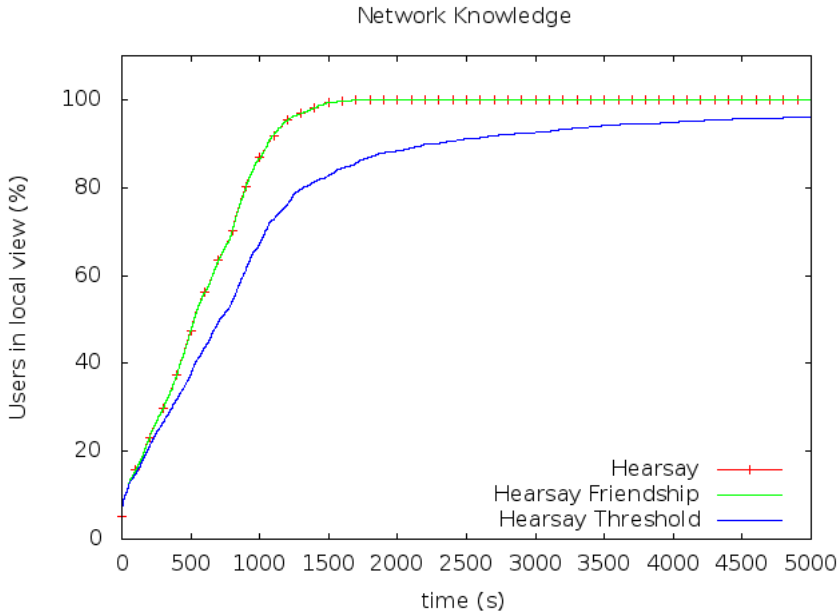


Figure 5.4: The average number of users in the social graph. 100% is reached when each user has knowledge about every user in the network.

5.2 Stability and Convergence

In a system without changes to the social graph, every user should have the same local view. Figure 5.4 shows a new scenario where all users were added to the network at the start of the simulation. Each user is allowed to add the first three strangers it interacts with as friends. Over time the relationship declarations will spread throughout the network until each node has received all social information. In the figure the average number of nodes in the local view is used to measure this. Hearsay Threshold is slower than the other two versions as it blocks friendship messages of users with a low rank.

A user leaving the network cannot be detected by the system as the communication is not guaranteed in a PSN. To remove old users from the network users should revoke their trust. Revoking trust uses the same message as giving trust, so the system will reach a stable state. Other options like limiting the lifetime of social graph information is also possible but that is part of the future work.

Ranking can also be measured for the stability of the system. As the local view stabilises the ranking of users becomes fixed. After starting the simulation the local view only contains the node itself and possibly a few friends. As more users are added to the local view, at greater social distances, the average ranking drops. The average rank of the system should stabilise if users do not receive changes to the network.

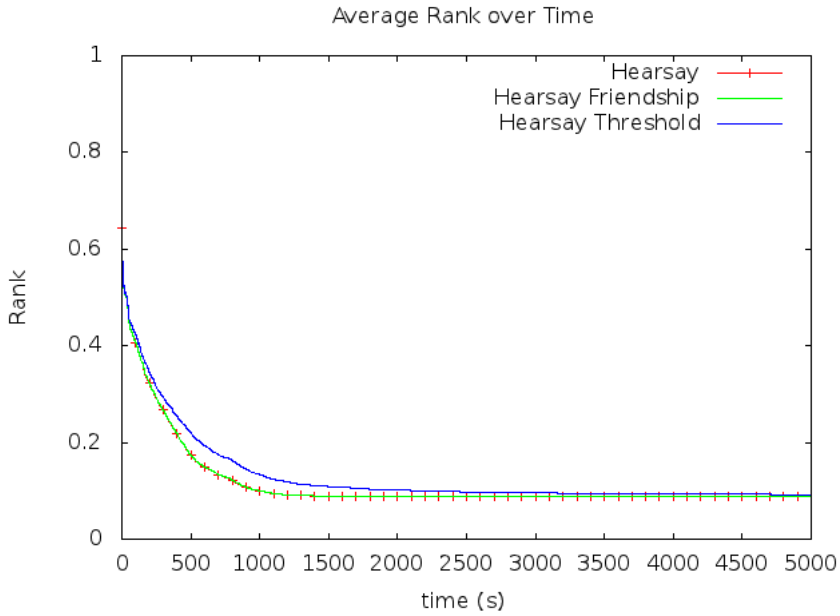


Figure 5.5: The average of all ranks of every user. Ranking has a steady decline as users learn more about other users with a high social distance.

$$avg = \frac{\sum_{i \in S} \sum_{j \in L_i} rank_i(j)}{\sum_{i \in S} |L_i|} \tag{5.1}$$

The average rank is calculated using Equation 5.1. In this equation S is the social graph and L_i the local view of user i, which includes user i and all other users known to him. $rank_i(j)$ is the rank that user i has assigned to user j. The average rank is the mean of every rank assigned by all users. If there are ten users this would be the mean of a hundred ranks as each users has ten ranks assigned. Figure 5.5 shows the average over time and it converges within an hour.

The previous simulations show that Hearsay is able to reach a steady state where each node has the latest information about the social graph.

5.3 Propagation Speed

The propagation speed depends on various parameters and this section illustrates what the effects of changing them are. A higher density increases the propagation speed, while a shorter range decreases it. Density and radio range have different influences, although both increase the number of neighbours. However, a larger radio range reduces the number of hops needed to cover a large area. Higher density just increases the number of users who

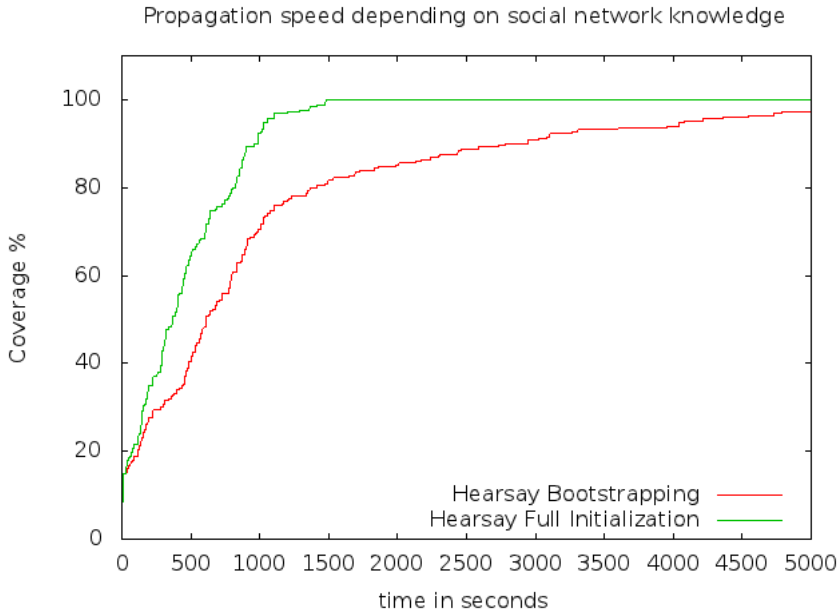


Figure 5.6: Propagation speeds for different methods of bootstrapping. Full Initialisation gives each user the complete social graph where every user has three friends, while with Bootstrapping each user has an empty local view in the beginning and adds friends during encounters.

receive the message and forward it. Over time the performance also changes as new messages are added to the network.

Messages are sorted according to their rank, which depends on who sent it and their relationship to the user. The ranking effects the speed at which a message propagates and is influenced by the number of messages, available communication time, and knowledge of the social graph. The effect of the number of messages was already illustrated in Figure 5.3.

The influence of knowledge about the social graph is harder to simulate, but not impossible. If a message is sent at the beginning of bootstrapping there is little knowledge of other users and their rank. At the end of bootstrapping all users should have determined the correct rank and message propagation happens according to the actual rank.

Figure 5.6 shows that messages sent without complete knowledge of the network propagate slower. The effect only causes a large delay compared to a network that started with a fully initialised social graph. The bootstrap version adds during the first few hundred seconds every node that come within range as a friend until the maximum amount is reached. After this period more strangers come within range, which slows the propagation. The social graph used for this simulation is a random social graph where each user links to a fixed number of other users, three in this simulation. Other

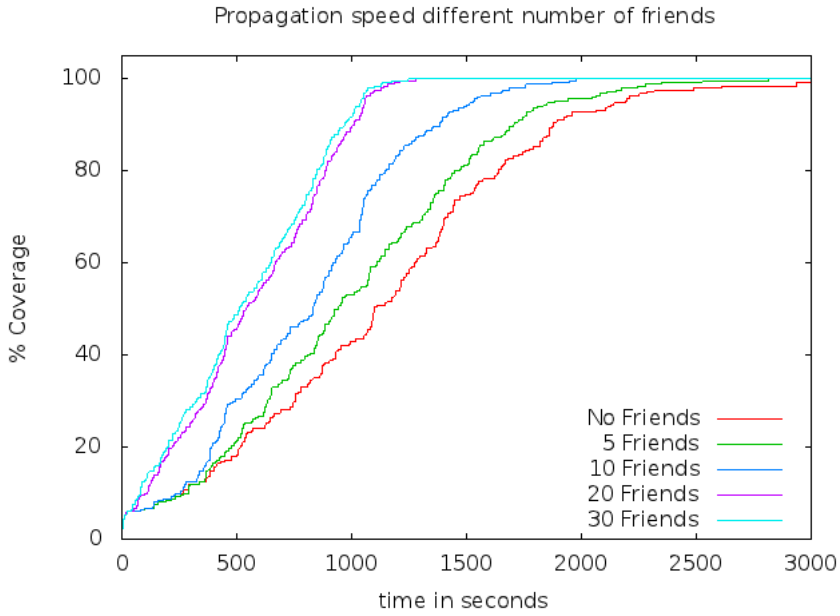


Figure 5.7: The number of friends influences the propagation speed as more friends are willing to spread the message.

social graph structures have different propagation speeds as nodes are more or less connected. In this thesis a study on different social graphs is not included, but it will be in our future work.

Not only the knowledge of the social graph determines how fast messages propagate but also the number of friends. This was discussed in Section 3.2.2 and the effects are shown in Figure 5.7. Each test created a random social graph for all users except one, who was assigned the required number of friends. After this initialisation period the selected user generates a message and its propagation is plotted in the figure. Two hundred messages were generated by the other users. During the test all users generate new messages to simulate usage of the network. The two tests with the highest number of friends give roughly the same high performance, which means that it is close to the maximum propagation speed for a gossiping network. The propagation speed with no friends is the lowest with a slight improvement for a users with five friends. Hearsay has no restriction in sending messages and therefore messages of the user with no friends can reach the entire network.

5.4 Spamming

Having analysed the general properties of Hearsay, we now turn to the main objective of this thesis, suppressing spam. To measure the effectiveness of suppressing spam we define a signal to noise ratio. Each user has a list

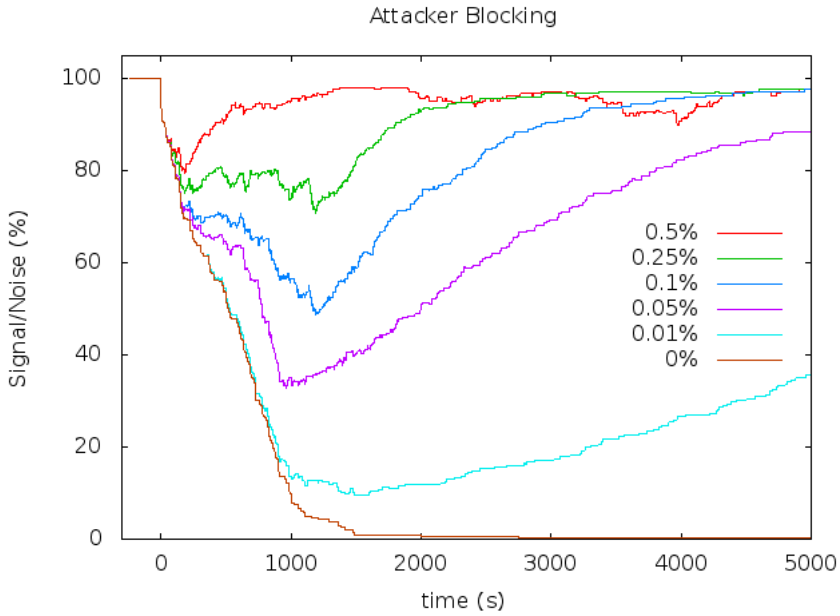


Figure 5.8: The effect of blocking probabilities, which is related to the time it takes for a user to block an attacker. Quicker blocking results in a better signal to noise ratio. However, blocking the attacker too fast will result in multiple small attacks as some users did not receive messages from the first spam wave.

of messages it is willing to send and it could contain spam messages. The signal to noise ratio is defined in equation 5.2 and is used to calculate the percentage of spam messages, M_{spam}, in the outbox, M, of a user.

$$\frac{|M| - |M_{spam}|}{|M|}, where\ M_{spam} \subseteq M \tag{5.2}$$

We simulate users blocking spammers manually by using a probability of 0.1% to block a spammer once a user has received spam originating from the spammer. Every second a user tries to block the spammer and on average a user blocks the spammer within 1,000 seconds. Figure 5.8 illustrates the choice for this probability. A lower probability ($< 0.1\%$) will have similar properties, such as a single *spam wave*, but increases the recovery time. A spam wave is a large drop in the average signal to noise ratio. A higher probability ($> 0.1\%$) blocks too quickly or suffers from multiple spam waves. Multiple spam waves happen if the first wave does not spread to all users. The remaining users have not seen any spam and therefore do not block the attacker. Once the attacker comes within range of these users a second spam wave will hit, visible in Figure 5.8 for the blocking probability of 0.5% and near 2500 and 4000 seconds. However, fast blocking is very effective even

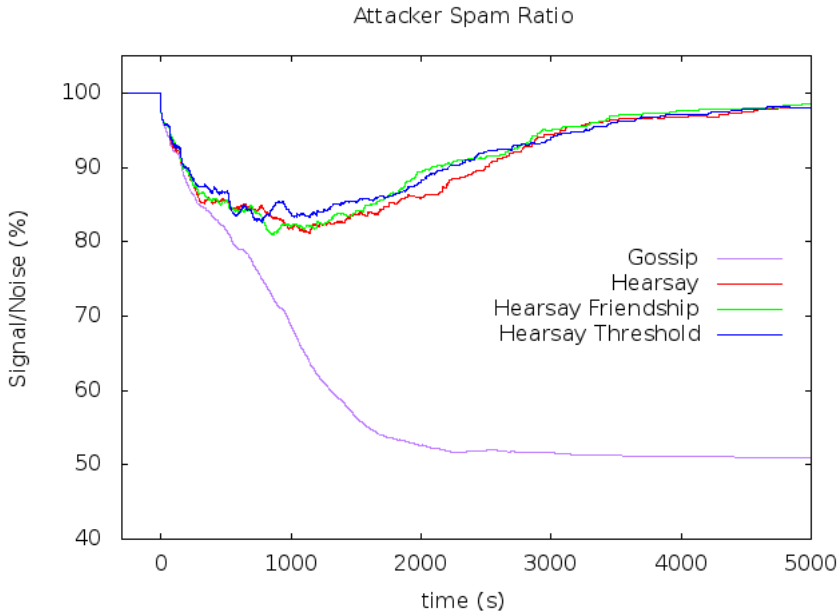


Figure 5.9: The effectiveness of spam suppression compared to gossiping. This clearly shows that gossiping is unable to block spam, while Hearsay is able to suppress spam. The network is under attack by a single spammer who is trusted by five users.

with multiple spam waves, only it takes longer before each user successfully blocks the spammer.

Figure 5.9 shows how effective Hearsay blocks an attacker compared to gossiping. The network has the same number of spam messages as normal messages and gossiping quickly converges to a signal to noise ratio of 50%. Hearsay blocks the attacker and distributes the removal of the friendship. The threshold and friendship versions of Hearsay are better at recovering from spam as users block the sending of spam if the attacker is outside of its local view. Threshold is the winner in limiting the effect of an attacker, while Hearsay Friendship has a steeper slope during recovery. The recovery speed depends on how fast the friendship messages are propagated. Friendship is faster than Hearsay because the latter also includes normal messages of unknown senders.

The simulation included a single attacker to demonstrate the spam suppression. More attackers will have a larger effect on the network as the amount of spam increases. However, if users keep blocking attackers the spam will eventually be suppressed similar to a single user.

An attacker with a lower rank will have a slower propagation of its messages. For efficient spamming an attacker should first infiltrate a group of users by gaining trust. With a higher rank the spam will spread faster and

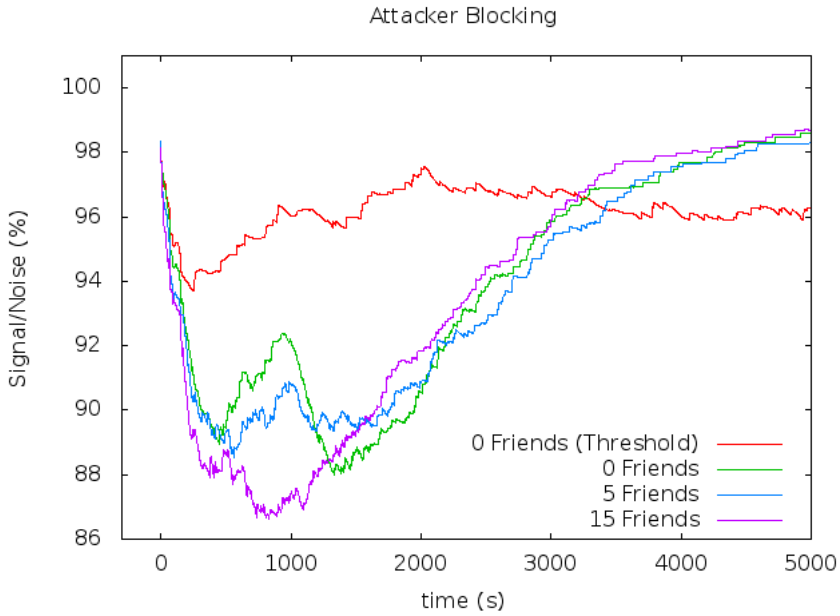


Figure 5.10: Attacker blocked by user as they register spam. More friends results in a faster spreading of spam, while fewer friends makes it easier to block an attacker. The network contains a single attacker that broadcasts spam.

reduce the propagation of other messages.

Such a situation is depicted in Figure 5.10. At the start of the graph the attacker reveals itself as such by sending spam. After this it has a small window to transmit large volumes of spam before other users react to it. As soon as the first users revoke the trust the signal to noise ratio returns slowly to normal.

The difference between the results are due to multiple properties of Hearsay. Having many friends results in the worst signal to noise ratio as many users contribute by forwarding the spam. However, having more friends results in a quicker recovery as many users transmit the removal of the friendship. Five friends performs as expected by having a slower recovery but the signal to noise ratio is better. Zero friends has a peak around 1000 seconds which can be explained by the lack of bandwidth. During the period before the peak there were other messages with a higher rank that needed to be forwarded. When users have transmitted all higher ranked messages a new spam wave starts.

Hearsay Threshold limits the influence a spammer has as users do not forward any spam messages and those received come directly form the attacker. Due to this intended behaviour the signal to noise ratio is around 96%. However, as no users forward the spam each users receives spam as it

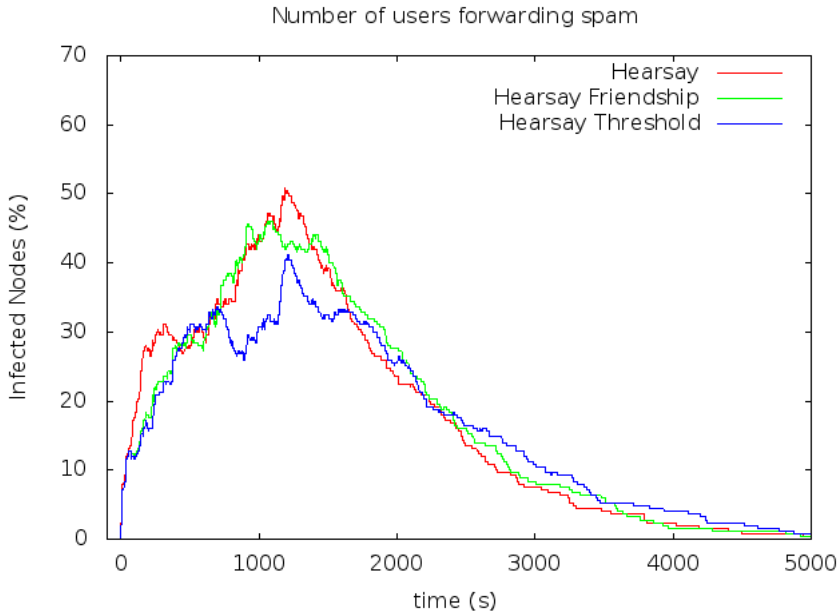


Figure 5.11: Number of infected nodes during the simulation of Figure 5.9. An infected node is a device that has received a message from an attacker and is willing to forward it.

interacts with the attacker. This is why the signal to noise ratio remains at the same level.

Infected users, or users who received spam or friendship messages of an attacker and forward it, can harm the network. Figure 5.11 illustrates how many nodes are infected during the attack simulated in Figure 5.9. The network consists of a single attacker with just five friends (10% of the users) but is able to infect 50% of the users. Hearsay does not limit the spreading of messages with a rank of zero and has the highest number of infected nodes. Hearsay Friendship starts similar to Threshold but is unable to keep the advantage. Compared to Hearsay it still performs slightly better as it spreads the friendship messages faster. Hearsay Threshold performs better in limiting the number of infected nodes as messages have a slower propagation speed, see Figure 5.2, providing the users an opportunity to block the attacker. However, Threshold limits the propagation of friendship messages slowing the recovery of nodes.

Figure 5.12 illustrates the propagation speed of messages during the simulated attack of the previous tests. Even with the added spam messages in the system the different versions of Hearsay have similar propagation speeds as achieved in the experiments shown in Figure 5.2. The differences between each version remains similar with Hearsay providing the fastest propagation. A direct comparison between a system with and without an attacker is dif-

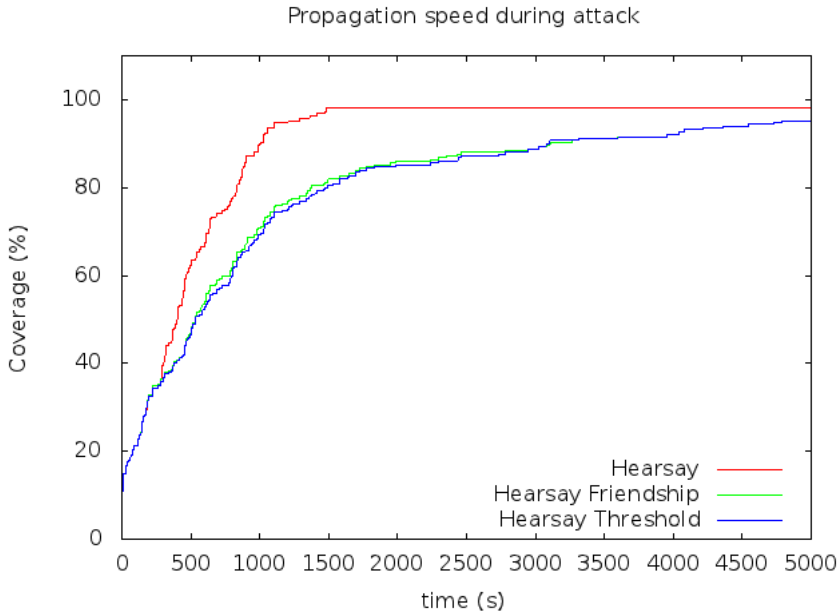


Figure 5.12: The propagation speed of a message, during a simulated attack, is similar to that illustrated in Figure 5.2

ficult to do as mobility patterns differ if more users are added to attack the network. However, the outcome can be predicted as a system under attack contains more messages slowing the propagation down.

Overall, Hearsay provides spam suppression while providing a robust network for communication. Figure 5.2 illustrates that the performance of Hearsay is similar to that of gossiping, while Figure 5.7 shows that the social network contributes to the propagation speed of messages. Finally, Figure 5.9 shows that Hearsay is able to suppress spam in a Pocket Switched Network.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The primary goal of this thesis is suppressing spam using trust in a social and mobile gossiping network. As described in the previous chapters the system relies on social information gathered opportunistically by the mobile phone of a user. All locally available information is combined to rank users and applied to order messages.

The design of Hearsay is based on TrustRank, a derivative of PageRank, which annotates the social graph with trust information. Based on the principles of trust, it established a trust score for users in the network. Common friends will gather more trust and new users will need to get trusted by others before they can send messages throughout the network.

Without relying on a centralised component or bootstrapping requirements Hearsay is usable in classic Pocket Switched Network scenarios such as a disaster. But the main focus was to build a network that could suppress spam and provide a robust communication network to trusted users. The end result is a system that can be deployed in a hostile and censorship-prone environment.

Simulations described in Chapter 5 clearly show an improvement over regular gossiping which has no spam suppression at all. Spam messages are limited to the local neighbourhood while the propagation speed of messages within the network is only marginally slower. An attacker who has infiltrated the network will be purged within an acceptable time after revealing itself as a spammer. This time depends on the willingness of other users, who trust the attacker, to remove their relationship.

As the system only uses locally available information about the social graph an attacker cannot easily spoof its rank. An attacker is required to create friendships with other users otherwise the spam is limited to the users

it directly communicates with. These properties combined allows Hearsay to achieve spam suppression and resilience against attacks.

6.2 Future Work

An actual implementation of Hearsay will be needed to verify all our assumptions and simulations. But a successful implementation requires many devices and a group of users that closely resembles a social graph. Current high-end mobile phones contain the needed technology, but these are not widely available to test friend-of-friend relationships. A plugin for Facebook or other social networks could be developed to test the ranking of messages based on a large social graph. The opportunistic exchange of messages is missing as well as the social pressure to limit ones friend list to the users he trusts.

Social graph calculations can be costly in terms of power consumption of the mobile phone. Storage of the social graph should not be a problem as current mobile phones have large internal storage and can be extended with SD cards or similar. Calculating the rank requires multiple matrix operations and research done for PageRank can be used to build an efficient version for mobile phones. Other techniques to limit the size of the graph can also be applied, such as removing users with a low rank that have not changed.

To overcome a problem with the delivery of old messages a notion of time can be added. Currently a message of a user is queued according to their rank. Messages from lower ranked users will not be forwarded if a high ranked user creates a lot of messages. A fair-use policy to restrict the number of messages sent by a single user can ensure that messages of lower ranked users are forwarded.

Hearsay Threshold and Friendship currently use a threshold of zero, while future versions could include a higher or dynamic threshold. A dynamic threshold can restrict low ranking users to send messages if an attack was detected. Higher thresholds will limit the propagation speed and the coverage, but it will improve the spam suppression as spammers need a higher rank to infect other users.

Bibliography

[1] Facebook press statistics. `https://www.facebook.com/press/info.php?statistics`, 9 2011. Visited: 12 September 2011.

[2] T. R. Andel and A. Yasinsac. On the credibility of manet simulations. *Computer*, 39(7):48–54, 2006.

[3] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks, 2001.

[4] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. 2005.

[5] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the 12th conference on USENIX Security Symposium-Volume 12*, pages 2–2, 2003.

[6] S. Buchegger and J. Y. Le Boudec. A robust reputation system for mobile ad-hoc networks. In *Proceedings of P2PEcon*, 2004.

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

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. *University of Cambridge, Computer Lab, Tech. Rep. UCAM-CL-TR-617, Feb*, 2005.

[9] T. M. Chen. Governments and the executive "internet kill switch". *IEEE Network*, page 2, 2011.

[10] Abdur Chowdhury. Global pulse. `http://blog.twitter.com/2011/06/global-pulse.html`, June 2011. Visited: 7 September 2011.

[11] S. Cottle. Media and the arab uprisings of 2011: Research notes. *Journalism*, 12(5):647–659, 2011.

[12] R. Deepa and S. Swamynathan. A secure and lightweight service discovery model for mobile ad hoc networks. In *Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09. International Conference on*, pages 526–530, dec. 2009.

[13] J. Douceur. The sybil attack. *Peer-to-peer Systems*, pages 251–260, 2002.

[14] A. Dunn. The arab spring: Revolution and shifting geopolitics: Unplugging a nation: State media strategy during egypt's january 25 uprising. *Fletcher F. World Aff.*, 35:15–145, 2011.

[15] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, pages 28–34, 2005.

[16] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald. When tcp breaks: Delay-and disruption-tolerant networking. *Internet Computing, IEEE*, 10(4):72–78, 2006.

[17] S. Garfinkel. *PGP: pretty good privacy*. O'Reilly Media, 1995.

[18] D. Gavidia. *Epidemic-Style Information Dissemination in Large-Scale Wireless Networks*. PhD thesis, Vrije Universiteit Amsterdam,, June 2009.

[19] D. Gavidia, G. P. Jesi, C. Gamage, and M. van Steen. Canning spam in wireless gossip networks. In *Wireless on Demand Network Systems and Services, 2007. WONS'07. Fourth Annual Conference on*, pages 30–37, 2007.

[20] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587, 2004.

[21] A. Heinemann, J. Kangasharju, and M. Muhlhauser. Opportunistic data dissemination using real-world user mobility traces. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1715–1720, 2008.

[22] M. Helft and D. Barboza. Google shuts china site in dispute over censorship. *New York Times, March*, 22, 2010.

[23] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot. Pocket switched networking: Challenges, feasibility and implementation issues. *Autonomic Communication*, pages 1–12, 2006.

[24] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251, 2005.

[25] A. Hussain, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110, 2003.

[26] A. Jain and P. K. Sagar. Various security attacks and trust based security architecture for manet. *Global Journal of Computer Science and Technology*, 10(14), 2010.

[27] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, 2007.

[28] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: Security challenges for the new paradigm. In *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pages 1–10, 2009.

[29] D. Kidston and T. Kunz. Towards network simulations credibility: Lessons from applying five key principles. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–6, 2008.

[30] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82, 2004.

[31] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *Proceedings of the second international workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 37–40, 2006.

[32] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: The current state and new simulation tools. *Mobile Computing and Communications Review*, 9(4):50–61, 2005.

[33] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: the incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(4):50–61, 2005.

[34] Andrew G. Miklas, Kiran K. Gollu, Kelvink. W. Chan, Krishna P. Gummadi, and Eyal De Lara. Exploiting social interactions in mobile systems. In *In UbiComp*, 2007.

[35] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A. K. Pietilainen, and C. Diot. Are you moved by your social network application? In *Proceedings of the first workshop on Online social networks*, pages 67–72, 2008.

[36] A. Mtibaa, M. May, C. Diot, and M. Ammar. Peoplerank: Social opportunistic forwarding. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5, 2010.

[37] M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):59–70, 2007.

[38] Mawloud Omar, Yacine Challal, and Abdelmadjid Bouabdallah. Reliable and fully distributed trust model for mobile ad hoc networks. *Computers and Security*, 28(3-4):199–214, 2009.

[39] Global Disaster Relief on Facebook. Japan earthquake and tsunami. `https://www.facebook.com/media/set/?set=a.10150119188069936.290029.250083749935`, March 2011. Visited: 7 September 2011.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[41] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks, 1997.

[42] Anna-kaisa Pietiläinen, George Varghese, Earl Oliver, Jason Lebrun, and Christophe Diot. Mobiclique: Middleware for mobile social networking.

[43] B. Preneel. A survey of recent developments in cryptographic algorithms for smart cards. *Computer Networks*, 51(9):2223–2233, 2007.

[44] J. M. Pujol, R. Sangüesa, and J. Delgado. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 467–474, 2002.

[45] J. L. Qiu. Virtual censorship in china: Keeping the gate between the cyberspaces. *International Journal of Communications Law and Policy*, 4:1–25, 1999.

[46] K. Ren, T. Li, Z. Wan, F. Bao, R. H. Deng, and K. Kim. Highly reliable trust establishment scheme in ad hoc networks. *Computer Networks*, 45(6):687–699, 2004.

[47] Elizabeth M. Royer and Chai-Keong Toh. A review of current routing protocols for ad-hoc mobile wireless networks.

[48] H. Samavati, B. T. Ladani, and H. Moodi. Amlet: Adaptive multi level trust framework for manets. In *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*, pages 152–157, feb. 2011.

[49] Boudewijn Schoon. Dispersy: Distributed permission system. 2010.

[50] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed around mobile users. *IFIP WONS*, 2006, 2006.

[51] S. Shah. Distributed twitter. 2009.

[52] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and trust-based systems for ad hoc and sensor networks, 2006.

[53] S. Trifunovic, F. Legendre, and C. Anastasiades. Social trust in opportunistic networks. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6, 2010.

[54] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.

[55] U Wilensky. Netlogo. `http://ccl.northwestern.edu/netlogo/`, 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

[56] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[57] Z. Yan and S. Holtmanns. Trust modeling and management: from social trust to digital trust. *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*, 2008.

[58] Kim Zetter. Undersea cables cut; 14 countries lose web — updated. `http://www.wired.com/threatlevel/2008/12/mediterranean-c/`, December 2008. Visited: 11 April 2012.

[59] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 506–515, 2008.

[60] C. N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4):337–358, 2005.