UNIVERSITY OF CALIFORNIA, SAN DIEGO

# Mobile Malware Propagation and Defense

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Gjergji Zyba

Committee in charge:

Professor Geoffrey M. Voelker, Chair
Professor Sujit Dey
Professor Bill Lin
Professor Stefan Savage
Professor Alex Snoeren

2013

The dissertation of Gjergji Zyba is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

# DEDICATION

To my mom, dad and brother.

## TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

Graduate school represents the most enlightening experience of my life; Not only because of the challenges that came with it, but also because of the great people that were involved in it.

First and foremost, I am extremely thankful to my advisor Geoff Voelker. He is a great teacher and mentor and I feel so lucky to have been his student. I especially thank him for being calm and supportive when things became overwhelming, for directing me to see the whole picture when I had the tendency to focus on less important details, for teaching me the art of a good presentation, for being a great listener and explainer, and for providing me with the tools to work through problems and challenges that initially seemed unsolvable.

I profusely thank my collaborators from Ericsson Research, Michael Liljenstam and András Méhes. They have literally been my other advisors and words cannot simply explain my gratitude for their dedication throughout the years of our work together. I thank them for their insightful suggestions and guidance, for providing me with all the necessary resources, without which my research would not have been possible, and for their close support, advising, communication and commitment. I especially thank Michael for all the frequent phone meetings, work and direct communication we had despite the time difference between Sweden and California, and for his constructive criticism and suggestions for improvement.

I am eternally grateful to Stratis Ioannidis and Christophe Diot, my internship supervisors at Technicolor Research Lab in Paris. I want to especially thank Stratis for his close support and communication and Christophe for his suggestions, directness and humorous attitude that made it so much fun; I promise I will try to retain my social vagabond status even after graduation.

"Thank you"s are simply not enough for James Mitchell and Eamonn O'Neill from University of Bath. Research is a collaborative effort and their contribution and insights have been invaluable. I especially thank James for his fun companionship during my stay at Bath.

I am indebted to Per Johansson from Calit2 for his insightful comments and suggestions during our early morning meetings, for opening up collaborations and providing me with necessary data for my research.

Nothing would have been possible without the right resources. I thank Brian Kantor, our SysNet administrator, for always being available and making computing and networking issues a non-issue, Jeffrey Cuenco for collecting and preparing the UCSD bluetooth sensor traces for our measurements, and Abderrahmen Mtibaa and Alloy Vincent for providing me with traces and computing resources at Technicolor Research Lab in Paris.

I thank my thesis committee members Sujit Dey, Bill Lin, Stefan Savage, and Alex C. Snoeren for their valuable feedback and support from proposal to completion of this effort.

I really feel the need to thank Julie Conner, our graduate student program director, for all her advice and support regarding paperworks from the moment I was admitted at UCSD until graduation. I also thank Jennifer Folkestad for patiently going through complex documents for all my academic travels.

My years as a graduate student would not have been so much fun without the companionship of all my close friends and colleagues.

I cannot think of my student life at UCSD without mentioning my friends Didem, Patrick, Ruomei, Amogh, Priti, Yuvraj, Gunjan, Ahmet, Tikir, Marisol, Nikos, Winnie, Panos, Konstantinos, Effie, Mary, and Levon for being who they are and spending countless memorable moments together. They have always been next to me for both the good and the hard times and I cannot express how happy I am to be surrounded by such loving, fun, and supportive friends.

I especially need to thank my colleagues Frank, Bhanu, Malveeka, Terry (UCSD), Vlad, Kshitiz (San Jose) and Ozan (Kista). Work is so much more fun with great colleagues and having to share the workspace with them is invaluable.

Last but not least, special thanks go to the people who have been next to me all my life and especially throughout grad school: my mom, Tatjana, dad, Jani, and my brother, Mirjan, for all their love and affection. I am extremely grateful to Sarah for always being a loving and supportive companion and her family Susan, Sande and Shelley for their support and enthusiasm towards my graduation.

Portions of this thesis are based on the papers which I have co-authored with others.

- Chapter 3, in part, is a reprint of the material as it appears in the “Proceedings of the IEEE Infocom Conference 2009” with the title “Defending Mobile Phones from Proximity Malware” by Gjergji Zyba, Geoffrey M. Voelker, Michael Liljenstam, András Méhes and Per Johansson. I was the primary investigator and author of this paper.

- Chapter 4, in part, is a reprint of the material as it appears in the “Proceedings of the IEEE Infocom Conference 2011” with the title “Dissemination in Opportunistic Mobile Ad-hoc Networks: the Power of the Crowd” by Gjergji Zyba, Geoffrey M. Voelker, Stratis Ioannidis, Christophe Diot. I was the primary investigator and author of this paper.

- Chapter 5 is based on the material as it partly appears in the “Proceedings of the 4th International Conference on Communication Systems and Networks 2012” with the title “Limitations of scanned human copresence encounters for modelling proximity-borne malware” by James Mitchell, Eamonn O’Neill, Gjergji Zyba, Geoffrey M. Voelker, Michael Liljenstam, András Méhes, Per Johansson.

# VITA

| | |
|---|---|
| 2006 | Bachelor of Science, Athens University of Economics and Business |
| 2009 | Master of Science, University of California, San Diego |
| 2013 | Doctor of Philosophy, University of California, San Diego |

# PUBLICATIONS

J. Mitchell, E. O’Neill, G. Zyba, G. M. Voelker, M. Liljenstam, A. Méhes, P. Johansson, “Limitations of scanned human copresence encounters for modelling proximity-borne malware“, *Proceedings of the 4th International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, 2012.

G. Zyba, G. M. Voelker, S. Ioannidis, C. Diot, “Dissemination in Opportunistic Mobile Ad-hoc Networks: the Power of the Crowd“, *Proceedings of the IEEE Infocom Conference*, Shanghai, China, 2011.

G. Zyba, G. M. Voelker, M. Liljenstam, A. Méhes, P. Johansson, “Defending Mobile Phones from Proximity Malware“, *Proceedings of the IEEE Infocom Conference*, Rio de Janeiro, Brazil, 2009.

ABSTRACT OF THE DISSERTATION

**Mobile Malware Propagation and Defense**

by

Gjergji Zyba

Doctor of Philosophy in Computer Science

University of California, San Diego, 2013

Professor Geoffrey M. Voelker, Chair

Over recent years, mobile devices, such as smartphones and tablets, have become feature-rich computing devices with networking opportunities that often surpass those of traditional PCs. Moreover, the smartphone market alone is now bigger than the PC market and, consequently, we see an exponential growth in the amount of mobile malware developed. Compared to traditional malware, mobile malware exhibits unique properties which require extensive studies to effectively protect the user. This dissertation identifies propagation vectors of mobile malware and examines characteristics of its propagation along with the effectiveness of various defense strategies.

I focus on the propagation of mobile malware when spread through direct pairwise communication mechanisms (e.g., Bluetooth). I evaluate, both theoretically and by simulation, the effect of user mobility on propagation, and find that malware can infect the entire susceptible population in days for a campus size area.

Proximity malware propagation is “invisible” to the network operator and defending against it is particularly challenging. I explore three defense strategies that span the spectrum from simple local detection to a globally coordinated defense. I find that local proximity-based dissemination of signatures can limit malware propagation, while the globally coordinated strategies that rely upon infrastructure within the mobile operator network can be even more effective.

Furthermore, I study the effect of user social behavior on malware propagation. In a particular area I identify frequent and transient visitors and compare propagation using either set or all devices. My analysis indicates that transient visitors, previously considered unimportant, play an important role in propagation.

Because direct pair-wise device encounters significantly impact proximity malware propagation, I study the strengths and limitations of deploying static scanners for inferring such encounters that are difficult to observe. By comparing direct and “virtual”-scanner-inferred encounters, I indicate significant statistical differences between the two categories, and find that malware propagation appears slower using inferred compared to actual encounters.

The results from our analyses give us a better understanding of the effect of different parameters in mobile malware propagation and defense against it. Our results also pinpoint limitations of using encounters inferred from static scanners for malware and, generally, any data dissemination.

# Chapter 1

# Introduction

During recent years, smartphone and tablet devices have become extremely popular due to the availability of feature-rich, custom applications. For the first time ever, smartphone sales outpaced PC sales for the year of 2011, marking a history-making trend [Can12]. At the same time, figures show that smartphone devices account for about half of the total number of mobile phones in the US [Nie12]. Not only do such devices provide a personal and portable computing environment, but they also come with always-on connectivity. As a result, these type of portable devices are increasingly becoming the target of malicious software, with over 17,000 variants of malware discovered by early 2012, marking a 22-fold increase compared to the previous year [Res12]. In this dissertation, I term this type of malware as *mobile malware*.

Many of the incentives behind traditional PC malware apply also for mobile malware. For example, mobile malware has been used for user information theft and credential exfiltration, novelty and amusement, search engine optimization and ransom [FFC$^+$11]. Moreover, the effects of user information and identity theft can be more severe on mobile devices due to their very personal character. Users store a variety of sensitive data (e.g., address book, calendar, SMS and email messages, personal photos and videos) in their mobile phones while they also carry the device with them most of the time. High-profile individuals have been the early targets of attackers trying to gain access to their sensitive data. In fact, an attacker that gets unauthorized access to the user's phone could not only retrieve such data but also locate (through GPS coordinates) and spy on the user (Internet activity, electronic transactions, eavesdropping). On the other hand, compared to traditional PC malware, mobile malware is frequently used for making money through premium calls and Short Message Service (SMS) messages, resulting in excessive fees for customersbringing back memories from the era of PSTN modems for PCs. Furthermore, the SMS creates a new vector for disseminating spam messages, irritating customers and mobile network operators. Mobile operators are also concerned as mobile malware poses a serious threat to their limited network resources. Operators want to preserve QoS and retain customer confidence, but relative to the Internet denial-of-service attacks require fewer resources to mount effectively [ETMP05].

The motivation for this work is that *understanding how mobile malware propagates is crucial in finding methods to contain it*. Smartphones, compared to PCs, have unique networking features that affect malware propagation. For example, a smartphone device is almost always carried by its user and it is in an always-on operating mode. Furthermore, at any time it may be active on a number of different networking interfaces (WiFi, Bluetooth, 3G/4G), providing multiple communication channels for data dissemination. Mobile malware has used Bluetooth to propagate [Labg, FS] and, more recently, the mobile operator network. In this thesis I focus on proximity malware propagation which fundamentally depends upon encounter patterns. By exploring these patterns I measure the dynamics of proximity malware propagation and evaluate defense mechanisms against it.

## 1.1 Contributions

This thesis addresses the question of what the dynamics of mobile malware propagation are, and how they differ when compared to the propagation of traditional PC malware. I aim to identify the different scenarios of mobile malware propagation, and further quantify them under different theoretical and realistic parameters. Furthermore, I present methods for containing mobile malware propagation and quantify their efficacy.

First, I studied the propagation of mobile malware that spreads through direct pair-wise communication mechanisms, such as Bluetooth or WiFi, between devices in geographic proximity. Our study evaluated the effect of user mobility on proximity malware propagation. For this study, we used both theoretical analysis and simulation along with different realistic mobility parameters. I showed that, even under conservative mobility parameter sets, malware can infect the entire susceptible population in a matter of a few days in a campus size area.

Second, I explored three strategies for detecting and mitigating proximity malware: *local detection*, in which devices detect when they become infected and disable further propagation, *proximity signature dissemination*, in which devices create content-based signatures of malware and disseminate them via proximity communication as well, and *broadcast signature dissemination*, in which a centralized server aggregates observations from individual devices, detects propagating malware, and broadcasts signatures to mobile devices. These strategies span the spectrum from simple local detection to a globally coordinated defense. I found that proximity signature dissemination can limit malware propagation to a fraction of the susceptible population, and do so without relying upon operator network infrastructure. The fraction of population that becomes infected (8% to 35% in our study) is highly dependent on the local detection thresholds. On the other hand, with the assistance of the mobile operator, broadcast dissemination is most effective, both because the rest of the susceptible population is immediately informed of the threat and the ability of the server to provide patches to the infected devices.

Third, I further studied the effect of user social behavior on proximity malware propagation. Social behavior dictates user mobility (e.g., areas someone visits the most, frequency and duration of visits), which in turn defines the encounters between particular individuals, which subsequently impact proximity malware propagation. In my study, I separated the population of a given area into two distinct groups based on their social status: “Vagabonds” and “Socials”. Vagabonds are devices that appear infrequently, whereas Socials are devices that appear in a repetitive/predictable manner. Most studies have considered Socials only for propagation and ignored Vagabonds because Vagabonds are considered unimportant (as they engage in fewer contacts) and encounter activity of such devices is limited in mobile device traces. However, I studied realistic traces of areas with different compositions of the two separate groups and I found that in typical scenarios Vagabonds consist of the majority of the devices. I further showed that, in addition to the social status, the number of devices in each group can play an important role. I showed both experimentally and analytically the “tipping” point beyond which the population size becomes more significant than the social status, predicting successfully when stealthy mobile malware spreads faster when using Vagabonds.

Fourth, I studied the strengths and limitations of deploying static scanners for inferring device-device encounters, which in turn directly affect proximity malware propagation. According to this method, devices which are detected simultaneously by a given scanner are considered to be spatially co-located (corresponding to observed “encounters” within the scanner’s radio range). I derived analytic results on errors introduced in scanner-based measurements for a simplified case where all scanners and devices are static. I examined the differences between device copresence as inferred by the scanners and actual copresence between the devices, and classified the discrepancies. Based on this classification, I then derived the probabilities with which each type of discrepancy will occur. Using simulation I validated our analytical finding that approximately 41% of copresence encounters inferred by scanners do not correspond to actual device copresence. Also using simulation, I demonstrated the extent and impact of errors when device mobility is included. I found that the set of encounters inferred from scanners differs from the actual encounters simulated in the model in terms of duration distribution and probability of encountering previously unmet devices. In all our simulation cases proximity malware showed slower propagation using scanned encounters compared to actual encounters for devices with the same mobility characteristics.

Taken together, this dissertation fills many gaps in our current knowledge of propagation dynamics of mobile malware. We now know to a better extent how fast malware can spread using pair-wise contacts, and to what degree the effectiveness of different (from local to global) defense mechanisms is. We also have a better understanding of the limitations of using contacts generated from static scanners for the purpose of malware and, consequently, any data dissemination. I believe these limitations should be considered by future research.

## 1.2 Organization

The remainder of this dissertation is organized as follows:

Chapter 2 covers background material useful for understanding the remainder of the dissertation and discusses related work.

Chapter 3 studies self-propagating mobile malware that spreads through direct pair-wise communication mechanisms. I show how long it takes for proximity malware to spread over the susceptible population in a university campus area. I do so by using synthetic generated contacts with properties similar to real contacts in captured traces, and I validate my results with an epidemiological analytical model. Furthermore, I present three distinct defense strategies against such malware and analyze their efficiency using our simulation framework.

Chapter 4 extends the work on proximity malware propagation with a thorough analysis of the role of human social behavior, in terms of mobility, on malware propagation. I show how a population in an area can be separated between regular/frequent and transitional visitors. By utilizing their encounters I show that, in addition to social status, the number of each group plays a significant role in virulent content dissemination.

Chapter 5 explores the benefits and limitations of using statically-located scanners to infer encounters between observed devices, a popular technique used by the research community when studying pair-wise communication, including the application of proximity malware propagation.

Finally, chapter 6.1 summarizes the work presented in this thesis and discusses future research directions in the field of mobile malware propagation and defense.

# Chapter 2

# Background

More than 30 years ago, *Elk Cloner*, the first personal computer virus that spread on the Apple II platform through floppy disks made its appearance. Since then, malware has evolved significantly and we have seen a lot of changes as personal computers became more popular, customizable and powerful. Seven years later the Morris worm was the first malware that would spread automatically, without user intervention, over the Internet. During the 1990s we saw a surge of macro viruses spreading through documents (e.g., Microsoft Word files) via email messages. The first virus that had the potential to damage hardware also appeared during that time [Labf]. The early 2000s saw the appearance of a number of viruses and worms that propagated extremely fast (*ILOVEYOU, Code Red, Nimbda, Blaster, SQL Slammer*) using networking software exploits and/or social engineering, and infected a large number of machines [MPS+03]. Also through exploits, a significant number of Web sites, including popular sites such as MySpace and Tom's Hardware, would be compromised, while some of them would forward malware to the users' computers who visited them.

Later in the first decade of the millennium computer malware increasingly became a tool for for-profit illegal activities. Infected machines from viruses and worms like *Sobig, MyDoom* and *Bagle* would be used to disseminate billions of spam messages [Reg]. Compromised machines would form botnets centrally controlled by a third party operator. Trojans, spyware and keyloggers would be used for stealing credit card numbers and user credentials, while rootkits would be employed to make malware non-detectable.

The history of malware for personal computers repeats on mobile phones, but in a much shorter time span. As smartphones become more and more powerful and popular, they also increasingly become a target for malware. The first instances of mobile malware date back to 2004. That year *Cabir*, a worm that propagated through Bluetooth between Symbian devices, made its first appearance [Labg] . Although Cabir was just a harmless proof-of-concept worm, later malware would use Cabir's propagating mechanism to spread. Around the same time, *Skuller* was the first malware that would use a system flaw and override system files on Symbian devices [Tro]. Its name comes from the fact that it was replacing OS icons with a skull and crossbones and disabling the device once it was restarted. Also during the same year, *Mosquit*, the first malware for financial gain, appeared [Labe]. It was a Trojan horse that would send SMS messages without user permission to premium numbers. Shortly after, *Commwarrior*, a worm that propagated using either Bluetooth or MMS messages, made its first appearance. By September 2005, Commwarrior was detected to have spread in more than 20 countries around the globe [Labb]. However, due to the limited target population (it infected certain Symbian devices) no major outbreak has been recorded.

Until 2009, the mobile malware scene remained mostly unchanged. Malware writers used Bluetooth as the main propagation vector of the malicious payload, while Symbian remained the main targeted smartphone platform, followed by Microsoft Windows Mobile. The smartphone market was increasing rapidly but still a very small fraction of the mobile phone market. In the meantime, cellular operators employed effective measurements against MMS propagating malware through content filtering [Labc]. The malicious users made most of their profit through Trojan horses which sent SMS messages to premium numbers without user permission. This malware was installed mainly through social engineering (e.g., luring WAP pages common for feature phones with Java 2 Platform, Micro Edition (J2ME) or Bluetooth content-sharing invitations) [Labd].

The introduction of the iPhone in 2008 and later the Android OS changed the mobile malware scene considerably. Currently, iOS and Android are the dominant smartphone platforms. Smartphones are easier to use, and provide a full-featured Web browsing experience. Furthermore, custom feature-rich applications are available for download through "app markets" managed by Apple and Google. Not surprisingly, mobile malware followed the trend.

*Ike* was the first worm for the iPhone. However, for it to propagate the device had to be jailbroken and the ssh password to be unchanged — two significant parameters that limited its propagation [Labd]. In fact, iOS devices can be effectively infected only through exploits, since the application market is very well controlled by Apple and users cannot install applications from unknown parties [McA]. This situation is the main reason why we have not seen a major malware outbreak for iPhones, even though they consist of a considerable fraction of the smartphone market.

However, this situation is not true for the Android platform which currently is the exclusive target for malware. Soon after the first Android phones were released, *FakePlayer*, an SMS Trojan horse, became one of the first malware apps targeting this platform [Labd]. FakePlayer's propagation was fairly simple: As soon as the user visited an adult content Web site and accepted the media player app to be installed, the device would become infected and immediately start sending premium rated SMS messages in the background. For FakePlayer to be installed, the user had to allow software installation from unknown sources and willingly download and install the app. The Trojan horse targeted Russian devices and did not become popular. Nevertheless, it was the first recorded malware for the Android platform.

The past two years have seen a tremendous increase in malware counts while malware complexity has also significantly increased. We have seen the appearance of botnet agents and backdoors for smartphones, spyware that steals user's data (from contacts, messages, phone calls and eavesdropping), Trojan horses that steal money (either through premium SMS messages, banking transactions, or credit card credentials) [Mob]. At the time of this dissertation, most of such malware employs social engineering along with the operator's IP network infrastructure to propagate (mostly through the Web). During this time, we also saw the first malware appearing on the Android application market itself (which lacks effective threat analysis [McA]) and as of January 2013, 99% of new malware targets the Android platform [Laba]. Pandora's box of malware seems to have finally opened!

The rest of this chapter outlines the current research literature on mobile malware propagation and defense through proximity mechanisms such as Bluetooth. Further related work on the impact of social behavior of users, in terms of mobility, on malware propagation is also presented, as well as previous research on inferring device encounters from real world traces.

## 2.1 Proximity malware propagation

Proximity malware propagation fundamentally depends upon user mobility dynamics. Data representing user mobility can be either collected from the real world or synthetically generated. Alternatively, analytic techniques can be used to model human interactions. Using either approach, a number of studies have demonstrated the threat of malware propagation on mobile phones through proximity vectors such as Bluetooth.

Specifically, Su et al. gathered Bluetooth scanner traces and used simulation to demonstrate that malware propagation via Bluetooth is viable, and explore its propagation dynamics [SCM$^+$06]. Kostakos et al. also simulated the propagation of proximity malware on extensive scanner traces of a downtown city center [Kos07]. The main drawback of using real world captured traces of user mobility for measuring malware propagation is that they represent a limited subset of devices and device-to-device encounters in an area. Additionally, different encounter assumptions can introduce errors which can confound certain statistical characteristics of human encounters.

Previous work on analytic models and malware propagation simulation using synthetic traces strived to accurately describe malware propagation over Bluetooth or other forms of proximity connections [YE07, MN05, RN08, ZLG06]. Yan et al. developed a detailed Bluetooth worm simulation [YE06] and analytic model [YE07], and showed that mobility can have a significant effect on propagation dynamics by simulating with different mobility models [YFC$^+$07]. The main issue with analytic models is that they do not capture spatial locality well; they can only roughly approximate mobility dynamics and are heavily dependent on the mobility assumptions [YFC$^+$07].

In this dissertation, I drew upon all three approaches (simulation with either real captured traces or synthetic traces, and analytic models) to inform our study. The Bluetooth scanner traces I gathered, for example, are similar to previous efforts [SCM$^+$06, Kos07]. I used simple epidemic models to verify the operation of our simulator and as a means for validating our simulation results. Going beyond simple random walk models [YE06], I used the Levy walk model for generating synthetic mobility traces; this model can describe the mobility patterns of a human being relatively well for concentrated areas [RSH$^+$08]. Recently, González et al. tracked mobile devices at large geographic scales (up to 1,000 km), showing that the Levy walk model is not as representative at those scales [GHB08]. However, I focused at much smaller scales, where spatial locality makes proximity malware most virulent.

## 2.2 Defending against proximity malware

Defending against proximity malware is challenging, mainly due to the fact that the propagation of such malware may not be visible through centralized networks (such as the mobile operator's network). As such, traditional network telescopes will fail to observe the propagation. Therefore, the only option is to start the defense at the device itself. Little work has previously focused on defense against proximity malware. Bose et al. and Kim et al. proposed two techniques for using behavioral signatures [BHSP08] and power signatures [KSS08] for locally detecting malware on mobile devices. Recently, antivirus applications have been available for the major mobile OS platforms. However, these applications need to be supplied with a set of signatures to be effective and may fail to detect a virulent dissemination of previously unseen malware instances such as zero-day exploits.

In this dissertation I investigated how such local detection impacts the overall propagation of malware in a population, particularly if devices actively coordinate to mitigate it. Therefore, our work is complimentary to the aforementioned local detection approaches in that they represent methods for deciding when a device has become infected (although malware may continue to propagate in the meantime). Thus, our primary goal is to understand the effectiveness of defenses, not to develop new mobility and modeling techniques.

## 2.3 User social behavior and proximity malware propagation

Proximity malware propagation is dependent on human encounters, which in turn depend on where, when, and how often people meet each other: their social behavior. For example, two coworkers will highly likely be in the same location for hours every day. Similarly, a teacher will meet her pupils on a daily basis. On the other hand, visitors to a university campus to give talks may have few local contacts before going back to the place they spend most of the time. In either of the above cases, the contacts between individuals are governed to a certain degree by their social habits and behavior.

Various studies have looked at the impact of social behavior on oportunistic mobile ad-hoc networks. Proximity malware propagation is effectively an application of message dissemination over temporary spacial ad-hoc connections.Routing protocols such as SimBet [HSL10] [DH07], Bubble Rap [HCY08] and PeopleRank [MMDA10] use social-based metrics derived from contacts between devices (such as betweenness centrality and neighborhood similarity) to make opportunistic forwarding decisions with low overhead. Protocols using explicit knowledge of friendship relationships have also been proposed and shown to improve efficiency over existing socially agnostic protocols [BRB$^+$08, MCL$^+$08]. All of the above protocols route over "strong ties" among mobile users, inferred either from contact behavior or declared friendships. In this dissertation, we extended these previous efforts by exploring the role and potential of devices carried by users that appear infrequently in an area (such as academic visitors above) for communication and data dissemination. Previous routing protocols ignored such devices (that I name Vagabonds) and, to the best of our knowledge, our work is the first to study their effect on virulent data dissemination.

Beyond routing, social networking concepts have been used in mobile opportunistic applications such as publish/subscribe systems [CMMP08, YHCC07], news feed [ICM09] and query propagation [MGC$^+$07, MSN05], and multicasting [GLZC09, POL$^+$09]. These systems make use of social networking concepts like node centrality [YHCC07], friendship relationships between nodes [MGC$^+$07, POL$^+$09, YHCC07], hotspots [MSN05], contact usefulness [CMMP08] and edge expansion [ICM09]. Our analysis focused mostly on epidemic message dissemination; nevertheless, our understanding of the effect of Vagabonds motivates further study of their effect on the behavior of applications like the ones described above.

## 2.4 Inferring device encounters for proximity malware propagation

Although several studies have considered propagation dynamics of, and defense strategies against, malware using proximity-based propagation [MN05, YHC07, SCM$^+$06, YE07, BHSP08], access to empirical data on which to base such studies is currently limited. What is required is either direct data on malware propagation, which is not generally available, or information about device encounters with which one can model propagation of malware.

In the absence of direct encounter data, human mobility data can also be used to infer encounters by considering spatial proximity between individuals. While human mobility data can be captured at a fine resolution under certain conditions, e.g., by using GPS traces, requiring users to record their movements is typically considered intrusive and onerous. As a result, studies that gathered such data had typically involved dozens or, at most, hundreds of users in a limited geographical area [KKK06, RSH$^+$08].

At the other end of the spectrum in terms of number of users and data granularity, recent work has utilized mobility data from mobile network operators (based on mobile phone and base station associations) [GHB08, SQBB10, ZB07]. This approach provides coarse mobility data for large numbers of users in a potentially large geographic area, but does not make it possible to determine when individual devices encounter each other and, for instance, are able to connect via Bluetooth.

Another option, when interested in Bluetooth connectivity, is to directly collect data by deploying an application on participants' Bluetooth devices which periodically scan for discoverable Bluetooth devices in range. However, as in the case of gathering GPS data, this approach is challenging in terms of both user effort and privacy. The Reality Mining project collected Bluetooth traces from approximately 100 users over a period of nine months [EP06] using an application installed on cellphones; however, the devices scanned only once every five minutes, which is likely to have led to shorter copresence encounters going undetected. In [CHC$^+$05], portable Bluetooth scanners were carried by up to twelve users for a period of five days, with scans being conducted every two minutes.

Some of the earliest papers on Bluetooth malware included empirical tests where Bluetooth devices were carried around to collect data on other discoverable Bluetooth devices encountered [SCM$^+$06, CMZ07]. In both of these studies, however, the volume of data gathered directly from devices was not sufficient to be used as a veridical source of encounter data for modeling a malware outbreak. Instead, Su et al. [SCM$^+$06] used the Reality Mining encounter data [EP06], while Carettoni et al. [CMZ07] used characteristics derived from their empirical data to parameterize a mobility trace generator based on social network theory [MM06].

Given the difficulty in gathering sufficient data directly from devices, other work has used fixed Bluetooth scanners to collect data on mobile Bluetooth devices passing by, including inference of what I call "copresence encounters" from the simultaneous presence of devices within the scanner's range [OKK$^+$06, KOP$^+$10]. The Bluetooth scanning approach enables the collection of data for large numbers of devices over long periods of time, but has its own limitations in studying proximity-based malware propagation. These issues have not been systematically examined in the literature and in this dissertation we explored the benefits as well as the limitations of this approach.

# Chapter 3

# Defense Against Proximity Malware

As we previously mentioned, mobile phones are increasingly becoming the target of malware. The potential effects of virulent malware propagation on consumers and mobile network operators are severe, including identity and information theft, permanently disabling devices (“bricking”), and excessive fees to customers or loss of revenue for mobile network operators.

As on the Internet, malware can use the mobile phone network to propagate. However, malware quickly encounters resource bottlenecks due to how mobile phone networks are provisioned. More importantly, with the ability to both centrally monitor and block propagating malware, mobile phone networks can quickly detect malware and employ defenses to contain it before it infects much of the susceptible device population [FLJ$^+$07].

Mobile malware, however, has another opportunity for propagation. It can propagate through direct pair-wise communication mechanisms, such as Bluetooth or WiFi, between devices in geographic proximity [Labg, FS]. Although slower than propagating over the network, *proximity malware* has the compelling advantage of being unobserved by the mobile network operator — making detecting proximity malware substantially more challenging.

In this chapter, we consider the dynamics of mobile phone malware that propagates by proximity contact, and we evaluate potential defenses against it. The dynamics of proximity propagation inherently depend upon the mobility dynamics of a user population in a given geographic region. Unfortunately, there is no ideal methodology for modeling user mobility. Traces of mobile user contacts reflect actual behavior, but they are difficult to generalize and only capture a subset of all contacts due to a lack of geographic coverage. Analytic epidemiological models are efficient to compute and scale well, but simplify many details. Synthetic models are flexible and provide the necessary geographic coverage, but lack the full authenticity of user mobility traces.

Lacking a single ideal method, we used all three approaches to study proximity malware propagation at the scale of a university campus. For evaluating detection and defense, we used the Levy walk synthetic mobility model because it provides complete geographic coverage of user contact dynamics, and previous work has shown it to be much more realistic than other synthetic models [RSH$^{+}$08]. We alsod use traces from Bluetooth scanners collected over seven months on the UCSD campus to corroborate the propagation results from the synthetic model. Finally, we used an analytic model as yet another method to cross-validate the baseline results of our simulator. Using multiple methods provides confidence in the implementation of our simulator and mobility modeling.

Defending against proximity malware is particularly challenging since it is difficult to piece together global dynamics from just pair-wise device interactions. Traditional network defenses depend upon observing aggregated network activity to detect correlated or anomalous behavior. With proximity malware, however, observations are inherently local since they do not involve network infrastructure. Proximity malware detection, therefore, must begin at the device.

We explored three strategies for detecting and mitigating proximity malware: *local detection*, in which devices detect when they become infected and disable further propagation; *proximity signature dissemination*, in which devices create content-based signatures of malware and disseminate them via proximity communication as well; and *broadcast signature dissemination*, in which a centralized server aggregates observations from individual devices, detects propagating malware, and broadcasts signatures to mobile devices. These strategies span the spectrum from simple local detection to a globally coordinated defense. We found that proximity signature dissemination can limit malware propagation to a fraction of the susceptible population, and does so without relying upon mobile network infrastructure. However, with the assistance of the mobile network operator, broadcast dissemination is most effective.

The rest of this chapter is organized as follows. Section 3.1 describes the infection models we considered, our simulation and analytic methods, and traces we used as input. Section 3.2 describes three defense strategies to proximity malware, and Section 3.3 evaluates their effectiveness on containing propagation. Finally, we summarize the findings of this work in Section 3.4.

## 3.1 Methodology

In this section we describe our methodology for studying proximity malware propagation. Fundamental to propagation is modeling how devices encounter each other as a basis for infection. We explore the use of three methods — traces of real user encounters, a synthetic model of user mobility, and an analytic model — and discuss how the methods can complement and corroborate each other, while also highlighting limitations in the process.

### 3.1.1 Simulating malware propagation

We simulated relatively simple, aggressive proximity malware. Such malware may exploit vulnerabilities in the communication stack [SCM$^+$06], or it may depend upon user interaction for successful infection [Labg]. Although relying upon user interaction may seem naive, such social networking "exploits" can be surprisingly effective. For example, the Cabir [Labg] worm periodically searches for discoverable devices via Bluetooth. If it finds a device, it asks to send a file using the OBEX protocol and waits for the remote user to accept. Cabir locks itself on that device until the victim accepts or goes out of range. In either case, it will search again for other devices. Commwarrior [FS] uses both MMS and Bluetooth to propagate. With Bluetooth, it searches for nearby discoverable devices as Cabir does. Instead of locking itself on one device, though, it asks multiple devices for a file transfer and sends to each device that accepts the file transfer a copy of the infectious file (as long as they remain within range).

We implemented a software simulator to model malware propagation and evaluate defense strategies. Our simulator accepts as input device encounters, i.e., events corresponding to when two devices start or stop being able to communicate with each other. We generated such events either from real traces or synthetic models. When two devices come within range of each other, the simulator adds them to a proximity list for each device. When devices move too far away to communicate, the simulator removes them from the lists. At any point in time, the simulator tracks whether devices are infected, communicating, disabled, etc.

The simulator models aggressive malware propagation. If an infected device is not busy (not currently communicating with another device), it will randomly pick up an idle device (not currently communicating) from its proximity list that it has not communicated with before. If it is unable to find a new idle device, it will remain idle until another device contacts it or is added to its proximity list. An infection takes some amount of time corresponding to delays due to Bluetooth association, file transfer, user interaction, malware execution time, etc. After this parameterized *infection latency*, the simulator changes the state of a vulnerable target to infected, and both devices are free to continue the propagation. The simulator also models a fraction $s$ of the mobile devices as *susceptible* targets of infection. Parameterizing susceptibility in a device population models such variabilities as hardware platform, software versioning, or user response, all of which may prevent malware from infecting a particular device. The simulator outputs the state of devices over time and the interactions that cause an infection. We use this output to compare different propagation scenarios and methodologies.

### 3.1.2 Real Traces

The primary advantage of trace data is that it reflects actual user behavior. Complementing other efforts [SCM$^+$06, Kos07], we collected trace data of user contacts using Bluetooth scanners installed at six active locations (the food court, main library entrance, etc.) on the UCSD campus. The scanners continuously scan for the presence of discoverable Bluetooth devices, sampling on average every 10 seconds, and store the scan results to a central database. Each record stores the unique ID for the device and the time that the device enters or leaves the proximity of a scanner; we assume that a device has left if it does not appear in a scanner scan list after 60 seconds. Over the span of seven months, these scanners recorded more than 1.6 million events for over 8,000 devices.

We transformed raw scan data into device contact events in two steps. First, we calculated the time intervals for which a device is within range of a scanner. Then, for each scanner, we determined all devices whose intervals overlap in time, and defined such devices to be in proximity contact with each other. As with previous studies, we made the assumption that two devices can communicate with each other if they are detected by the same scanner. Note that this assumption may include devices that may not actually be able to directly communicate since the actual locations of the devices is not known relative to the scanner position; for malware propagation, this assumption tends to be conservative since it may provide more opportunities for propagation than in actuality.

Since malware will take time to propagate between two devices, we keep track of the durations for which two devices are in contact. The simulator tracks malware propagation time to model whether the contact duration of two devices is long enough for malware to successfully propagate between them.

### 3.1.3 Synthetic Traces

One limitation with using real traces is that it is difficult to cover an entire geographic area with scanners. As a result, real traces will not capture device contacts outside of scanner range, potentially influencing propagation results. An alternative is to use a synthetic mobility model to produce device contacts for an entire geographic area, albeit an area simplified from any particular real location.

We used the Levy walk mobility model to generate synthetic device contact events. Previous work has shown that the Levy walk model can describe the mobility patterns of a human being relatively well for a campus-sized area [RSH$^{+}$08] (although perhaps not at larger scales [GHB08]). A Levy walk is a collection of flights, angles, and wait times. A flight is a straight line of movement of a given distance. An angle denotes the turning angle at the end of the flight. A wait time is the waiting time between two subsequent flights. Flight length and wait time are calculated from the Levy distribution with some scale factor $c$ (10 for flight length and 1 for wait time) and two parameters $a$ and $b$.

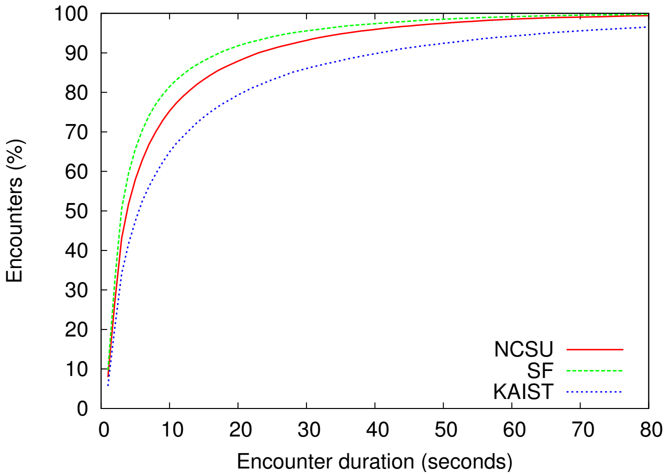
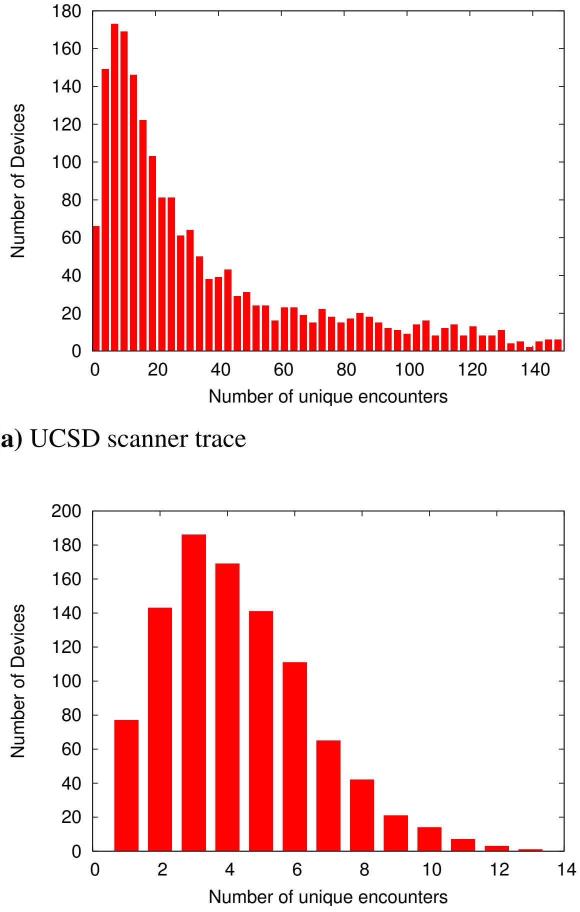


**Figure 3.1:** CDF of encounter durations using the Levy walk mobility model.

Parameters $a$ and $b$ affect the flight length (and speed, consequently) and wait time distributions, respectively. Rhee et al. derive different parameters for various user traces collected across a wide range of settings, from university campuses to amusement parks [RSH$^+$08]. The parameter values typically range between 0.5–1.5 and reflect the geographic characteristics of the place; some places tend to have more "localized" movements than others. Note that smaller values reflect longer flight lengths and wait times. From their experiments, for example, the "San Francisco" (SF) traces $(a = 0.75, b = 1.68)$ give the highest contact rate, "KAIST" traces $(a = 0.97, b = 0.45)$ the lowest, and the "NCSU" traces $(a = 0.86, b = 0.99)$ strike a balance between the two. A contact between two devices happens when the devices come within 10 meter range of each other (a reasonable distance for Bluetooth).

Recall that the simulator takes into account the infection latency, or the time it takes one device to infect another. For a successful infection, the devices must be in range long enough for a transfer to take place. Figure 3.1 shows the distribution of contact durations for the "SF", "NCSU", and "KAIST" parameterizations of the Levy walk model. Unless otherwise specified, in our experiments we used an infection latency of 10 seconds. Although this value might seem high at first, Bluetooth handshakes take at least five seconds, and transferring malware files the size of Cabir and Commwarrior double that time. As a result, infection latency substantially reduces the number of



**(a)** UCSD scanner trace

**(b)** Virtual scanners with Levy walk device mobility

**Figure 3.2:** Distributions of unique device contacts: $y$ devices encountered $x$ other unique devices.

infection opportunities in practice. For the “NCSU” Levy parameters, for example, only 25.4% of all encounters have sufficient duration to permit an infection.

### 3.1.4 Analytic Epidemic Models

To verify the operation of our simulator and to provide an additional means of comparison, we also employed a simple analytical SI epidemic model [DG99]. This model and enhancements have been usefully applied to malware propagation on the Internet, and we applied it to the problem of proximity malware as well. The model has the form:

$$\frac{ds(t)}{dt} = -\beta s(t)i(t) \tag{3.1}$$

$$\frac{di(t)}{dt} = \beta s(t)i(t) \tag{3.2}$$

where, at time $t$, $s(t)$ is the number of susceptible individuals, $i(t)$ is the number of infected individuals, and the total population is constant, so that $\forall t, s(t) + i(t) = N$. $\beta$ is the infection parameter (describing disease infectiousness).

This epidemic model, using differential equations, is an approximation of the average behavior for large populations of what is in actuality a stochastic process. The model assumes *homogeneous mixing* and is based on the law of mass action formulation. Homogeneous mixing means, roughly, that at any given time any two individuals have an equal probability to have a contact. The law of mass action, in a population context, implies that the rate of contacts between populations is proportional to the product of the sizes of the populations.

To calculate the infection parameter $\beta$, we considered the device encounters used to drive our simulations and compute the mean encounter rate $r$ over all devices. If all contacts are infectious, $\beta$ is the pairwise contact rate, i.e., $\beta = r_{xy}$ for devices $x$ and $y$. In practice, because a minimum contact time is required for Bluetooth to transmit the infection, we let

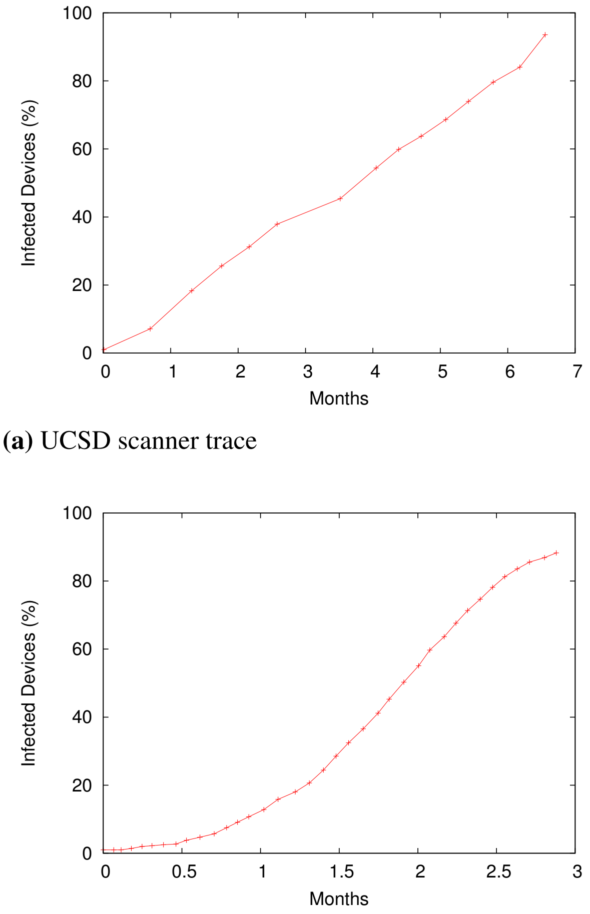
$$\beta = p_1 \cdot r_{xy} = p_1 \cdot \frac{r}{\binom{N}{2}}$$

where $p_1$ denotes the fraction of contacts long enough for a single (infectious) transmission.

Given the mean encounter rate $r$, the device population $N$, and the fraction $p_1$ as inputs, it is straightforward to numerically calculate the number of infections over time.

### 3.1.5 Corroboration and validation

Finally, we explored ways in which we can use these three methods of generating device encounters to complement and corroborate each other. For studying proximity



**(a)** UCSD scanner trace

**(b)** Virtual scanners with Levy walk device mobility

**Figure 3.3:** Malware propagation over time based upon contacts from real scanner traces (UCSD) and simulated virtual scanners (Levy walk).

malware detection and defense in Section 3.3, we primarily used synthetic contacts generated by the Levy walk model. The traces and analytic model are helpful, however, for corroborating results when using a synthetic mobility model and validating the implementation of our simulator.

### Bridging real traces and synthetic models

Consider the UCSD Bluetooth traces. They represent a finite number of scanners placed at specific locations in a larger geographic region. We can perform a similar experiment with the Levy walk model, randomly placing “virtual” scanners in a region and recording the contacts only observed at those virtual scanners. We can then compare the real trace and synthetic model from similar perspectives. We performed this experiment for the Levy walk model for 1,000 devices in a 9 km$^2$ campus-sized geographic region with six scanners. We used the "NCSU" parameters [RSH$^+$08] for generating Levy walks, which strike a balance between the "SF" and "KAIST" extremes. For each simulation run, we generated a Levy trace and randomly choose 1% of the devices to initiate the malware propagation. We repeated this experiment 100 times, choosing different initial devices on each run.

First, we compared the contact distributions from the UCSD trace and the virtual scanner traces from the Levy walk model. The contact distribution is an important metric reflecting the mobility characteristics of users in an area, particularly as it applies to propagating malware. Figure 3.2 shows the contact distributions for the UCSD and Levy traces. Each graph shows the number of devices *y* that encountered *x* other unique devices; e.g., in the UCSD trace 170 devices had 10 encounters with unique devices. As with previous studies of contact, we saw that a few devices are very popular while many have far fewer contacts. Although we do not expect the absolute parameters of the distributions to be the same for the two approaches, it is encouraging that this popularity property holds for both synthetic and real traces.

Next, we also simulated the propagation of malware using both contact traces. We simulated a very simple scenario where all devices are vulnerable and infecting a device on contact is instantaneous. We use the same percentage of initial infected devices as above. For this experiment, Figure 3.3 shows the percentage of infected devices as a function of time for the UCSD and Levy traces, respectively. In both figures, each point shows the median percentage across all of the different cases of initial devices. We do not expect these infection results to be good predictors of an actual infection due to their limited geographic coverage, which results in unrealistically long infection times (months!). However, in terms of evaluating the characteristic behavior of the underlying processes, we again found it encouraging that the curves exhibit roughly similar behavior — albeit with some variation, particularly at the ends of the distribution.

We considered these experiments a form of sanity check. Keeping that perspective in mind, though, we found it encouraging that the synthetic model, when used under similar constraints as real-world traces, produces roughly similar behavior in contact distributions and malware propagation as the scanner data. Note, however, that the results in Figure 3.3b reflect the use of "virtual" scanners, which only sample a spacial subset of an entire region, and will therefore differ from results that model all of the device contacts for an entire region (which we used when evaluating defenses in Section 3.3). Using contacts based solely on real scanner traces can predict malware propagation orders of magnitude slower than contacts based on an entire region, a limitation that needs to be kept in mind when using scanner traces as the basis for proximity malware propagation.

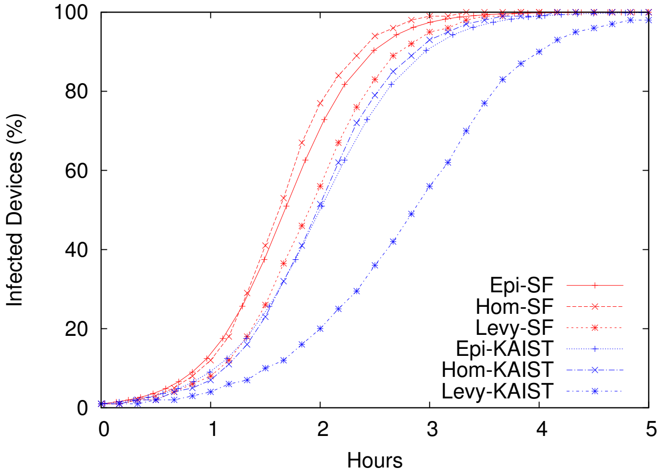


**Figure 3.4:** Malware propagation according to the epidemic model, homogeneous mixing contacts, and the Levy walk model.

### Bridging the analytic and synthetic models

We also used the analytic model in validating the implementation of the simulator. The simple SI model assumes homogeneous mixing. We generated a synthetic trace of homogeneous random contacts at a constant rate, and compared malware propagation based on the SI model and simulation using the homogeneous trace.

Figure 3.4 shows the results of a numerical solution of the epidemic model ("Epi") and results from two kinds of simulations, one using the "homogeneous mixing" of devices in an area ("Hom") and the other using the Levy walk model ("Levy").

The graph shows the percentage of vulnerable devices infected as a function of time. For each method, we show curves for two contact rates, the "San Francisco" (SF) and "KAIST" parameter sets described in Section 3.1.3; these two sets represent the extremes in contact rates among the traces evaluated in [RSH$^+$08]. For both the homogeneous mixing and Levy walk scenarios, we simulated a simple infection where all devices are vulnerable, and infection latency is 10 seconds.

Comparing the results of the epidemic model and simulation using homogeneous mixing contacts, we found that they are in close agreement, validating that the simulator operates as expected. Of course, homogeneous mixing contacts are not a good approximation for encounters of mobile devices. Previous work [MN05, YFC$^+$07] has shown that simulations of proximity malware propagation using random walks or random waypoint mobility generally do not appear to match results from the simple epidemic model, as the contact structure tends to differ from homogeneous mixing. However, as these mobility models are not considered particularly realistic, it was not clear what the implications are for comparing the epidemic model with more realistic mobility models such as Levy walks.

Comparing the results of the epidemic model and simulation using the Levy walk model provides insight on this question. For the SF scenario, with higher mobility, there is a small discrepancy between the simulations and the epidemic model, such that Levy walk contacts result in somewhat slower spread of the malware. Intuitively this slower propagation is not surprising, as one would expect spatial locality in the contacts resulting from Levy walks that is not present in a homogeneous mixing model. The spatial locality implies that initially only contacts occurring in a subregion around the initial infection location can transmit the infection. Hence, this behavior slows down the spread of the malware. For the "KAIST" scenario, with shorter flight lengths and longer pause times resulting in less "mixing", this discrepancy is more pronounced.

## 3.2 Proximity Malware Defense

We evaluated a set of defenses under the following general scenario. We assume that devices have a trusted defense software component that can examine messages and files transferred between devices, securely record persistent information about these transfers, and control device hardware when necessary (e.g., disable radio communication). These assumptions may be strong, but not unreasonable given the increasing prevalence of trusted computing modules. However, if malware has the ability to disable defense software, we can predict what the result will be: unchecked propagation through a population, as in Figure 3.3 and previous work.

We modeled malware detection at the device using one or multiple *contacts* with other infected devices as evidence. This approach idealizes the use of a particular detection technique. For example, a device may be able to detect that it has become infected on first contact with another infected device using a behavioral technique like system call anomaly detection. Or the device may have received a signature, from another device or the network, that enables it to use content analysis to detect on first transmission that a message or file is malware.

Alternately, a device can be suspicious about being contacted by multiple other devices to transfer the same file, eventually concluding that the other devices are likely infected and the file they are sharing is malware. Having multiple independent devices try to communicate the same file is reasonably suspicious behavior, and certainly characteristic of propagating malware. A paranoid setting might label a file as malware after only two such contacts; a more lenient, perhaps more risky, setting might wait for three or more contacts. Such a technique requires storing hashes of file content managed as a cache, and is vulnerable to polymorphic attacks.

Note that, for any technique that requires more than one contact, after the first contact we assume the device is infected and can infect other devices. Also, any of these techniques may lead to false positives. Without good workload studies on communicating Bluetooth applications, however, it is difficult to evaluate what the false positive rate of any particular technique may be. This issue remains an open question.

Based upon this general scenario, we explored the following three defense strategies. We abstracted any specific technique by parameterizing the number of contacts with infected devices required for a device to decide whether it has become infected.

### 3.2.1 Local detection

The first strategy simply uses local evidence to detect malware and prevent further dissemination by the device, such as by disabling the Bluetooth or WiFi radio. Preventing further propagation by disabling communication may inconvenience the user, but voice and messaging with the mobile network remain possible. Disabling the malware prevents further propagation but makes no attempt to notify other devices or the network about the presence of malware. It serves as a useful baseline for comparison. We parameterized local detection with a threshold number of contacts $n$ with other devices: after $n$ unique contacts, the device decides that it is infected and stops propagating the malware.

### 3.2.2 Proximity signature dissemination

The second strategy extends local detection with an active mitigation component. In this strategy, each device maintains a table $S$ of signatures of malware files, such as an MD5 hash over the file content. After a device $X$ infers that it is infected, it disables[1] the malware and warns subsequent devices about it. Device $X$ computes a content-based signature $s$ over the file(s) that triggered the infection recognition (e.g., the hash it has used to track file transfers in the first place). When $X$ comes into proximity contact with another device $Y$, $X$ disseminates the signature $s$ to $Y$. If $Y$ is infected, it immediately disables the malware. $Y$ then adds $s$ to its signature table $S$. Whenever another device shares a file with $Y$, $Y$ will check the file against the signatures in $S$. The device can then either delete the file, or warn the user about the file.

Note that this strategy makes a number of assumptions. It is, in effect, a "white worm" [WE06]. Such worm defenses have not been effective in the Internet environment, but it remains an open question for proximity malware. It also assumes that a device can trust signatures received from other devices. Using signed signature transfers between the defense software components on both devices would be one mechanism for establishing this trust.

[1] For example, the trusted component may start filtering outgoing infection attempts.

### 3.2.3 Broadcast signature dissemination

The third strategy relies upon the mobile network operator to disseminate signatures using a broadcast mechanism. In addition to standard unicast messaging, mobile network operators are also able to send data packets over broadcast at low cost [3GP].

In this strategy, whenever a device decides that it is infected, it sends the malware content to an anti-virus server in the mobile network (e.g., using MMS). The server, since it presumably contains far greater processing power than the mobile devices, can compute a better quality signature. Also, due to access to anti-virus experts, the server may also be able to compute a patch that contains information on how devices may "cure" themselves, i.e., remove the infection from the device. Manual involvement in generating patches is also a possibility.

Then, when the mobile network operator's anti-virus server receives at least $m$ alerts from unique devices for a particular malware instance, it broadcasts a generated signature to the entire mobile network. This signature immediately cures all infected devices and renders uninfected devices invulnerable.

## 3.3 Experimental results

In this section we evaluate the three mitigation strategies described above. We implemented them in our simulation framework and used the Levy walk mobility model as the basis for generating device contacts. Unless otherwise specified, we simulated 100 devices in a 1km$^2$ region with one initially infected device, and used ten random Levy walk traces with 100 random initial device selections each. Since it is tractable to extend the epidemic model to capture the effects of the local detection strategy, we present results for that method as well for comparison.

### 3.3.1 Local Detection

Local detection removes infected devices from the population by disabling them, but otherwise does not actively try to prevent malware propagation. Figure 3.5 shows that local detection does little to prevent proximity malware propagation. For a variety of experiments, the graph shows the percentage of devices that are infected over time. We assume that all of the devices are susceptible; Section 3.3.4 relaxes this assumption.

The "Levy" curves show results from simulations with the Levy walk model. The "No mitigation" curve shows the unrestrained propagation of malware. Malware propagation infects over 80% of devices in just over 2 hours. The "T=*n*" curves simulate local detection with a threshold of *n* contacts (e.g., a device decides that it has been infected if it has been contacted by *n* other unique devices transferring the same file to it, and it sees itself also transmitting that file). An aggressive threshold of two delays propagation by another 3 hours. Increasing the threshold to three substantially reduces the effectiveness, only delaying propagation by about 30 minutes. Higher thresholds have little impact whatsoever. The curves for a threshold of four (not shown) nearly overlap the "No mitigation" curve; we simulated for higher threshold values with the same effect.

These results use the "NCSU" Levy walk parameterization. We also simulated the effects using the other parameterizations. For this and other results below, the effects are similar to those shown in Figure 3.4: parameters corresponding to higher contact rates ("SF") propagate malware more aggressively (shift the curves left), and those with lower rates ("KAIST") do the opposite. Because the results do not fundamentally change with different Levy parameters, we continue to show results for the "NCSU" parameters.

As a point of comparison, we also present results using the analytic epidemic model (Section 3.1.4). We extended it to model device local detection and shutdown as follows. Once a device disables itself it enters the removed state *R*, and transitions to the removed state occur as the number of received infection messages reaches the contact threshold. Typically, we expect this transition to happen in an encounter between two already infected devices.

During experimentation with this model, however, we found an unexpected effect to be significant: depending on the behavior of the malware, it is possible that an infected device gets removed in the course of an encounter with an uninfected device. If the encounter lasts long enough, and the malware aggressively tries to propagate itself (with high frequency and without keeping track of which devices it has come from) the

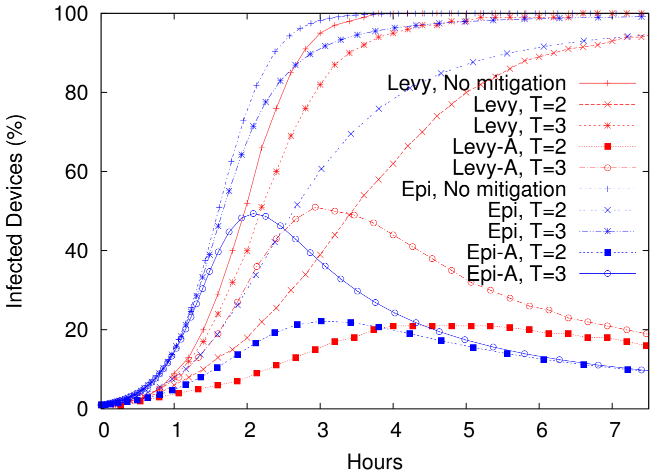


**Figure 3.5:** Effects of local detection on malware propagation, evaluated both with Levy walk simulations and the epidemic model.

newly infected device may contact the already infected device. Thus, in a quick "reflected infection" exchange, the previously infected device receives an infection message. Thus, for a small threshold value like two, a single encounter with an uninfected device could lead to shut-off. We let $p_2$ be the probability of an encounter between an infected and a susceptible device resulting in such a "reflected infection" exchange. Hence, for a threshold value of 2:

$$\frac{ds(t)}{dt} = -\beta s(t)i(t) \tag{3.3}$$

$$\frac{di(t)}{dt} = \beta i(t)\left[(1-p_2)s(t)-i(t)\right] \tag{3.4}$$

$$\frac{dr(t)}{dt} = \beta i(t)\left[p_2 s(t)+i(t)\right] \tag{3.5}$$

where $r(t)$ is the number of removed devices at time $t$.
The model can also be extended to higher threshold values in a similar fashion. For a given infection latency, we determine $p_2$ according to the fraction of encounters whose duration meets the infection latency threshold (Figure 3.1).

The "Epi" curves in Figure 3.5 showed the results of the epidemic model when modeling local detection. As seen earlier (Section 3.1.5), the epidemic model tends to predict somewhat faster infection propagation than the results based on the Levy walk simulation, but are otherwise in agreement with those results. In particular, it is in agreement with the simulation in terms of the general trend: that some effect can be achieved using a threshold value of 2, but this diminishes quickly as the threshold is increased.

The curves denoted by “A” show the fraction of devices *actively* propagating the infection, i.e., infected and not disabled. Even though the dynamics of the two models differ somewhat, they predict approximately the same peak fraction of simultaneously active infected devices.

### 3.3.2 Proximity Signature Dissemination

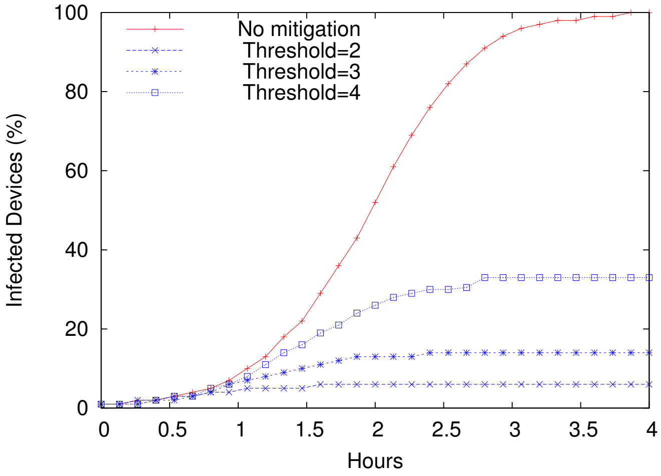


**Figure 3.6:** Infection over time using proximity-based signature dissemination.

With proximity signature dissemination, infected devices generate malware signatures and disseminate them via proximity contacts as well. Figure 3.6 shows the simulation results of this strategy for various thresholds using the Levy walk model to generate contacts. The results show that proximity signature dissemination does limit propagation: the strategy contains the infection to just a subset of the population. As expected, aggressive (lower) thresholds have a more significant impact than conservative (higher) thresholds.

An advantage of this strategy is that it is done entirely with the individual devices themselves; no anti-virus service for signature creation and dissemination takes place.

Given that no infrastructure by the mobile network operator is needed, we consider this mitigation strategy to be very effective. With relatively few Bluetooth flooding applications being used, for example, we argue that an aggressive threshold of two or three would be reasonable in the short term — in the current environment, being contacted to transfer the same file from multiple independent devices is suspicious. However, as Bluetooth applications evolve, the threshold question would likely have to be revisited.

### 3.3.3 Broadcast Signature Dissemination

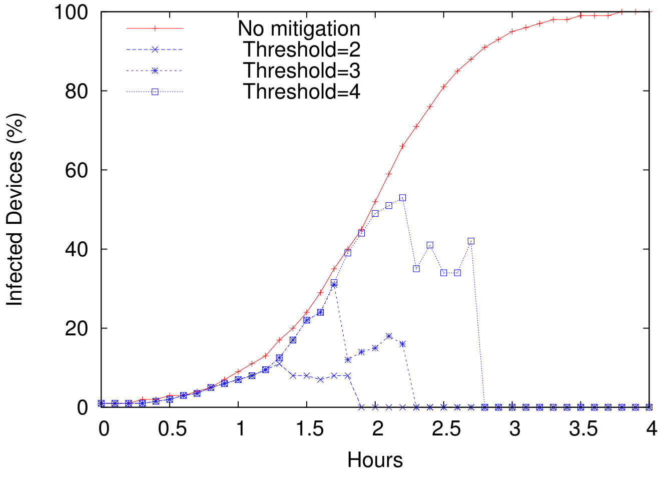


**Figure 3.7:** Infection over time using broadcast-based signature dissemination.

With broadcast signature dissemination, when devices decide that they have become infected they send the malware files to a central server, nominally controlled by the network operator. Once a sufficient number of devices $m$ contact the server about the same malware instance, the server — either automatically or with manual help from experts — generates a patch that can disable the malware, and broadcasts it to devices. Figure 3.7 shows the simulation results of this strategy for various contact thresholds, again using the Levy walk model to generate contacts. We modeled a central server that triggers a broadcast after $m = 20$ devices contact it. Note that, since devices can be patched during the propagation, the percentage of infected devices can decrease over time.

The results show that, under these assumptions, broadcast dissemination is highly effective at not just limiting, but also removing malware from the network. While this strategy does require the mobile network operator to invest resources and take action (e.g., maintenance of anti-virus server, possibly manual generation of recover-enabled signatures by anti-virus experts), an increase in malware incidents could motivate the cost. An alternative model is that operators could label it a premium service and charge customers, perhaps along similar business models as current anti-virus vendors.

### 3.3.4 Susceptibility

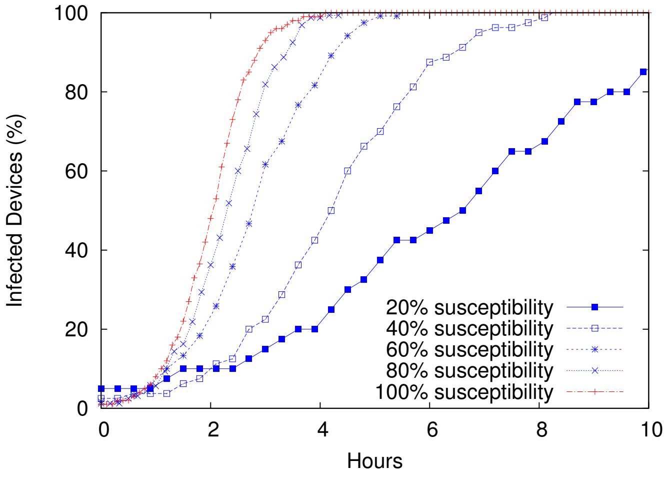


**Figure 3.8:** Malware propagation with varying degrees of susceptible populations. Percentage of infected devices is over the susceptible population.

Our previous experiments have assumed that all devices in the population are susceptible. In our final experiment, we relaxed this assumption by varying the percentage $s$ of susceptible devices in the population. As discussed in Section 3.1.1, parameterizing $s$ can represent hardware, software, or user features that determine malware susceptibility. For instance, malware that exploits a vulnerability in Symbian, independent of device, would have a value of $s$ corresponding to the prevalence of Symbian phones in a population. Alternately, for malware that requires user interaction to propagate (e.g., accept a file transfer), $s$ can reflect the fraction of users willing to accept such transfers.

Figure 3.8 shows the effects of varying *s* between 20–100% of the population. Note that we calculated the percentage of infected devices relative to the susceptible population. The results show that malware still effectively infects the entire susceptible population, but that reducing *s* increases the time it takes for complete propagation. The effects become more pronounced for smaller values of *s*. At 40% and 20%, malware takes increasingly longer to propagate through the population as the density of susceptible devices decreases substantially.

## 3.4 Summary

Proximity malware spreads through ad-hoc communication wireless technologies such as Bluetooth. We used the Levy walk mobility model to investigate the propagation of a hypothetical worm using the proximity vector. We validated the results with both a basic epidemic analytic model and real traces that we collected at the UCSD campus. As we saw, under realistic assumptions, this kind of malware can propagate quickly in small and populated areas.

The challenging threat of proximity malware is that it can propagate among mobile devices without being observed by the network operator. This behavior makes the propagation of proximity malware difficult to detect and mitigate. In this chapter we evaluated three strategies to address this problem. These strategies use multiple contacts with other infected devices as evidence to determine locally that malware is propagating. A baseline strategy that performs local detection and containment has only marginal impact on propagation. A more active strategy that combines local detection with proximity dissemination of signatures has a much more dramatic effect, limiting malware propagation to only a fraction of the susceptible population. Finally, a global strategy relies upon the network operator to coordinate local observations and generate patches to not only contain the malware, but also to patch devices through broadcast communication.

Chapter 3, in part, is a reprint of the material as it appears in the “Proceedings of the IEEE Infocom Conference 2009” with the title “Defending Mobile Phones from Proximity Malware” by Gjergji Zyba, Geoffrey M. Voelker, Michael Liljenstam, András

Méhes and Per Johansson. The dissertation author was the primary investigator and author of this paper.

# Chapter 4

# The Effect of User Social Behavior on Proximity Malware Propagation

As we saw in the previous chapter proximity malware can use opportunistic encounters between devices that happen to be co-located. Proximity malware propagation is essentially a “data dissemination application” in an opportunistic ad-hoc network. Independently of what technology they rely on, opportunistic mobile ad-hoc networks allow communication in a natural and effective way, taking advantage of locality and mobility to increase information exchange opportunities. However, in general, communication in such opportunistic mobile ad-hoc networks is challenging due to the volatility of contacts, communication technologies, and resource limitations (e.g., batteries, communication opportunities, wireless data transmission technologies). Communication is also strongly impacted by human mobility, which is driven by user social behavior.

Despite substantial work in the area, both theoretical and experimental, our understanding of these networks is limited. Progress in understanding opportunistic mobile ad-hoc networks is mainly limited by the difficulty to collect complete traces, and to model large systems with realistic assumptions (which is linked to the absence of large experimental data sets). The main difficulty in the experimental approach is to collect traces that (i) contain enough information about each device (in particular its mobility, social profile of its owner, *exhaustive* list of contact opportunities, duration of contacts and communication technology impact) and (ii) are not biased by constraints due to experimental conditions.

In particular, there is a need to collect and consider data that encompasses the behavior of all devices in a population—not just experimental devices—to have a complete view of the experimental environment. Indeed, most data sets collect information in a predefined experimental population, such as participants carrying GPS receivers [RSLC07], Bluetooth scanners [CHC$^+$06] and smartphones [EP06], and WiFi PDAs [MV05]. These data sets have at best a partial view of the environment, and of the role non-experimental devices could play in malware dissemination. This situation is best illustrated by the Hong Kong trace explored in [HCY08] where the experimental devices have strictly no direct contact with each other, yet they contact thousands of external devices that could play an important role in malware dissemination but for whom it is not possible to collect data.

In this chapter, we use publicly available traces to improve the understanding of virulent message dissemination in opportunistic mobile ad-hoc networks. We overcome the limitations identified above by choosing traces that collect information about all devices in an area (and not only a limited set of experimental devices). We further process these traces by subdividing each trace based on a specific social or professional geographical area of interest. We observe that a significant amount of devices appear rarely within a given area, and, because of their large population, we explore their impact on virulent message dissemination. In each sub-trace, we define two classes of populations with different presence characteristics, namely *Socials* and *Vagabonds*. Socials are individuals who return periodically to a specific area (analogous to the experimental devices in the discussion above, or to community members). Vagabonds instead are seen more rarely and randomly (i.e., the external devices that are in general not measured, or removed from traces because of partial information). A device can be a Vagabond in one area, and Social in another as well as change its role over time, thus exhibiting both spacial and temporal characteristics.

The first contribution of our work is to study, for the first time, malware dissemination spanning a large range of Social and Vagabond compositions. Previously, most studies consider Socials only and ignore Vagabonds entirely, or have just a partial knowledge of them because of experimental conditions.

Second, we observe that the efficiency of content propagation is not only a consequence of the devices' social status, but also a consequence of the number and density of devices. We see that in many cases, due to their large population, Vagabonds are more effective in spreading a message, even though they are considered unimportant. They therefore play a key role in malware dissemination and they should not be ignored. This result contrasts previous works that focused only on the effect of social properties on dissemination [HCY08, MMDA10, HSL10].

Third, we study both experimentally and analytically the "tipping" point beyond which the population size becomes more significant than the social status. We do so by observing this behavior on our traces but also by developing an analytic model that formally characterizes the relationship between population size and the social behavior of users. Our analysis confirms our experimental results and identifies a simple formula for determining when malware dissemination through Vagabonds outperforms dissemination through Socials.

Section 4.1 describes the data sets we use in this study. Section 4.2 introduces three possible definitions of the Social and Vagabond groups, and analyzes their properties in each area. Using the most promising definition, we study the mobility characteristics of Socials and Vagabonds in Section 4.3. Then we analyze the impact and role of each group on content propagation using trace-driven simulations in Section 4.4. Finally, we formulate an analytical model that captures Social and Vagabond mobility properties to explain and extend our results in Section 4.5, and conclude in Section 4.6.

## 4.1 Data Sets

We use traces from three data sets.[1] We specifically chose these traces because they represent distinct and considerably different mobile environments. We avoid using traces of experimental devices only (e.g., participants in a conference) unless all existing devices (even the ones not seen by the experimental devices) are monitored. We refer to these data sets as *Dartmouth*, *San Francisco (SF)* and *Second Life (SL)*, according to the location where they were collected. We further subdivide Dartmouth and SF into smaller geographical areas which have different social behavior characteristics. Table 4.1 sum-

[1] Two available through CRAWDAD at http://crawdad.cs.dartmouth.edu

marizes the basic characteristics of the three data sets we consider. We discuss below the features of each data set and our motivation for using them.

**Table 4.1:** Basic characteristics of the data sets: population size, trace length, type of area, population type and logging frequency. The population size is the number of devices that have at least one contact with another device.

| **Data Set** | **Pop.** | **Length** | **Area** | **Pop. type** | **Log Freq.** |
|---|---|---|---|---|---|
| San Francisco | 483 | 24 days | City | Cabs | 1–3 mins |
| Dartmouth | 4248 | 60 days | Campus | Devices | Instant |
| Second Life | 2713 | 10 days | Small | Avatars | 1–3 mins |

**Dartmouth** The Dartmouth data set comprises logs of association and disassociation events between wireless devices and access points at Dartmouth College [HKA08]. The logs span 60 days and include events from 4920 devices. Of these, 4248 have at least one contact with another device, and we focus our study on these devices. As with many previous studies using WiFi traces (e.g., [CHC$^+$06, MV05]), we assume that two devices are "in contact" when associated with the same access point.

We identify three areas within the Dartmouth campus likely visited by different social communities, shown in Figure 4.1: *Engineering* (300m×200m), and *Medical* (300m×300m) are specific schools while *Dining* (150m×150m) corresponds to the main food court of the Dartmouth College campus where we expect all students to mix. The main features of this data set are that (1) it logs *all* WiFi devices on campus, as opposed to only preselected experimental devices in prior work [HSL10, HCY08, MMDA10], and (2) each region represents different social behavior in a university environment. However, the assumption that contacts take place between any two devices associated to the same access point may introduce a bias compared to real contact opportunities (as shown in Chapter 5).

**San Francisco** The San Francisco data set consists of GPS coordinates of 483 cabs operating in the San Francisco area [PSDG09], collected over a period of three consecutive weeks. We assume that any two cabs can communicate whenever their distance is

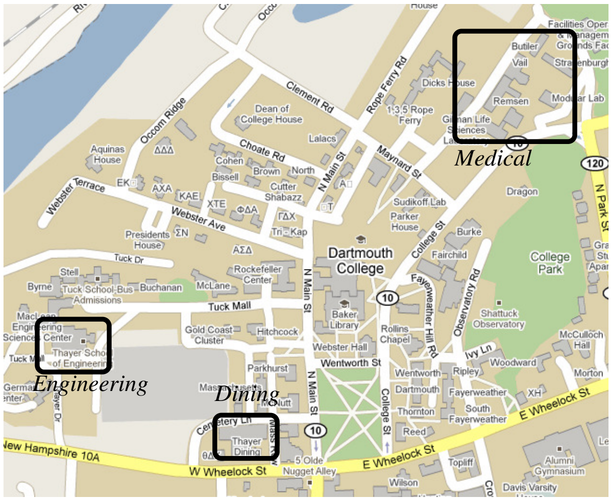


**Figure 4.1:** Dartmouth College map. We further subdivide our trace by focusing on three sub-areas surrounding the buildings housing (a) the School of Engineering, (b) the Medical School and (c) the main campus Dining food court.

less than 100 meters, a realistic range for WiFi transmissions.[2] We select three regions of San Francisco in which we expect cabs to exhibit different behavior. We refer to these areas as *Sunset* (2km×6km), *Airport* (0.7km×1km), and *Downtown* (2km×2km).

Our cab population is not exhaustive but represents all vehicles in a cab company comprising a large proportion of the San Francisco cabs, which number around 1500 [SFM09]. The interest of this trace is that it represents the behavior of taxi drivers in different parts of a city where some of them live, park their cab, or simply decide to wait for customers because of their friends or social habits. Their social behavior is clearly impacted (and possibly dominated) by customer requests and the lack of information about customers is clearly a limitation of this trace. Nonetheless, the SF trace is very interesting as it is representative of a community behavior across the different areas we study, and it is the only environment where the ratio of Social and Vagabond varies significantly. Last, it is worth noting that mobility in this trace is mostly defined by traffic conditions and speed limits.

[2] We tried other values and observed no significant difference for ranges of 100–300 meters in our results.

**Second Life** The last data set captures avatar mobility in the Second Life (SL) virtual world [Sec, VPDB08]. The data set consists of the virtual coordinates of all 3126 avatars that visit a virtual region during 10 days. We assume that two avatars are able to contact each other and exchange data when they are within a vicinity of 10 meters, a reasonable range for close-proximity communication such as Bluetooth [PD09]. The number of avatars which engage in contacts is 2713, and as with the Dartmouth trace we study only these avatars. It has recently been shown that the social network defined by such contacts between SL avatars resembles real-life social networks [VV10].

We do not define sub-areas in this data set as the virtual region is too small (300m$\times$300m). This limitation is balanced by the exhaustive user population captured, where Socials are people returning on regular basis and Vagabonds are occasional visitors that come only once in most cases.

## 4.2 Socials and Vagabonds

We first classify users according to their social mobility behavior. To do so, we divide the user population in each trace into two distinct groups: *Socials* and *Vagabonds*. Intuitively, Socials are the devices that appear regularly—and, therefore, predictably—in a given area. In contrast, Vagabonds are devices that visit an area rarely and unpredictably.

Based on the above intuitive definition, we propose three different methods for classifying users into Vagabonds and Socials, and we apply these methods to the selected areas of the three data sets we presented in the previous section. By definition, the classification of a user as a Social or a Vagabond will depend on the area one considers. For example, it is possible that a user is a Social in the Engineering area of Dartmouth, and a Vagabond in the Medical area.

### 4.2.1 Identifying Vagabonds and Socials

The first method classifies users based on how long they stay in a given area. The other two methods classify users based on the regularity of their appearance in an area.

The results shown in this section focus on a five-day consecutive weekday period, as we expect Vagabonds and Socials to exhibit different behavior between weekdays and weekends. We have verified, however, that our definitions and behavioral properties hold on all other five-day weekday periods in all traces.

### Least Total Appearance

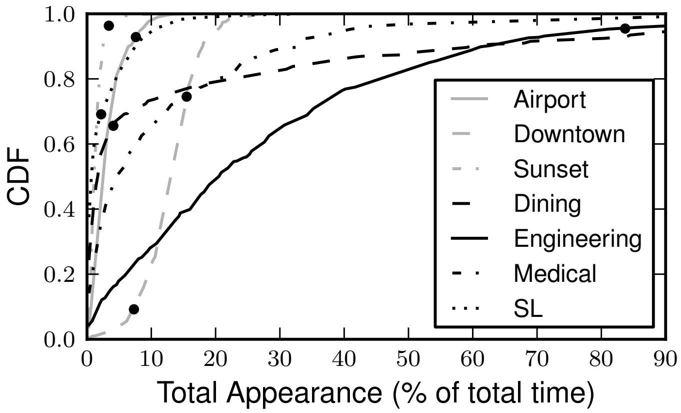


**Figure 4.2:** Total time the population appears in each area. The black dots represent the knees of the CDF curves as found by the linear regression method.

We define total appearance as the total time spent by each device within an area during the five-day period. Figure 4.2 shows the CDF of the total appearance time of the population for the first week of each area. In almost all areas (excluding Engineering) more than 75% of the population appears less than 20% of the time, with even lower appearance time being the common case. Thus, few devices stay within an area for longer periods and, intuitively, such devices would be the Socials of this particular area.

We define the least total appearance (LTA) threshold as the first inflection point ("knee") of the CDF of the total appearance for an area. This threshold separates Vagabonds from Socials, and is specific to each area.

To objectively identify such inflection points in the CDF curves of Figure 4.2, we employ a technique for detecting significant changes in curvature [Ley09]. Each curve is iteratively approximated by a straight line using linear regression in the range $[0,t]$, where $t > 0$. The iteration stops when there is a significant error in the approximation.

We assume that there is a significant error in approximation when the correlation coefficient $r$ is such that $r^2 < 0.9$. This point identifies the knee in the CDF, and thus also signifies the threshold that we should use in the LTA method.

Figure 4.2 also shows the inflection points for the different sub-areas as dots on their respective CDF curves. Although Sunset has a single clear "knee", Downtown and Engineering do not. Downtown has two possible inflection points, and LTA selects the lowermost. Engineering has no distinctly apparent knee. Its curvature varies slowly across the full distribution, and LTA eventually selects a point as the CDF levels off.

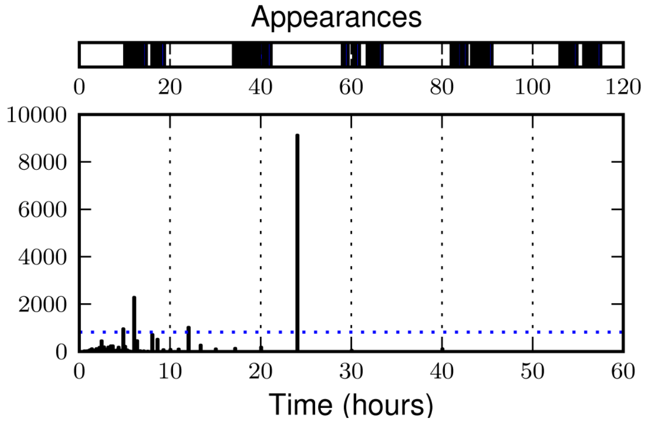


**Figure 4.3:** An example of a social device detected using the Fourier method.

**Fourier**

Our second classification method, *Fourier*, detects periodicity. It relies upon the Fourier transformation and the autocorrelation of the appearance of a user in an area, approaches used in signal processing to detect periodicity.

We employ a technique by Vlachos et al. [VYC05], and Figure 4.3 shows an example of applying this technique to a device in Dartmouth Medical. The top graph shows the appearance of the device throughout a five-day period. The next graph shows the Fourier transform of this signal into the frequency domain.

The Vlachos technique determines a threshold on the frequency coefficients in the Fourier transform. If the transform has coefficients above the threshold, the device appearance is periodic and corresponds to a social user. Otherwise, the device is a Vagabond. The bottom graph shows the threshold for the example device with a horizontal dashed line. Several Fourier coefficients exceed this threshold, and hence the device is Social.

For social devices, the technique identifies the inverse of the highest frequency coefficient as a potential period of the device appearance. The technique subsequently uses autocorrelation to improve the accuracy of the period estimate.

The Fourier method is problematic for nodes that appear very infrequently (e.g., once or twice). The spectrum of such nodes would be roughly uniform (e.g., white noise), making the selection of an appropriate threshold difficult. Consequently, almost half of devices that appear once or twice in certain areas were labeled as Socials by this method, which is clearly a mischaracterization. As a result, we investigate an additional method that focuses on periodicity.

### Bin

Our third method, termed *Bin*, is motivated by the observation that people's mobility patterns exhibit a diurnal behavior [GHB08]. Our traces also confirm this behavior, as the most frequent period detected by the Fourier method was 24 hours. Based on this observation, Bin detects if a user appears *every day* in an area, and *consistently* during the same time period.

For each trace we divide our measurement period into bins of equal size $b$, corresponding to the length of the "time period" during which a user frequents the area. We then represent the appearance of each device over time as a binary string, where each bit corresponds to a time bin. For each device, we flag a time bin with "1" if the device appears in the area during the period corresponding to this bin, and "0" otherwise.

We then consider a device to be periodic if it appears every day, at a specific period of the day. For a given bin size $b$, a device whose corresponding string has a "1" every $\frac{24}{b}$ bits is periodic. For flexibility, we identify a device as periodic even when an exact bin is not flagged but a neighboring (either previous or next) bin is. If a device is "periodic" by this definition we consider it Social, otherwise it is a Vagabond.

In experiments using the Bin method in this chapter, we use bin sizes of 3 hours. We believe that this size is representative of the time variance of the diurnal behavior of users from one day to the next. We obtained very similar results when repeating the experiments with a bin size of 4 hours, suggesting that around this time granularity the results are not very sensitive to the bin size.

### 4.2.2 Classifying Vagabonds and Socials

Table 4.2 shows the percentage of Vagabonds in each area according to each classification method. We observe that, under all methods, in most of the areas Vagabonds represent the majority of the population. The Downtown area in SF is an obvious exception: as expected, most cabs visit the downtown area frequently enough to be characterized as Socials by all three methods.

**Table 4.2:** Percentage of Vagabond devices in the areas.

| Area | Total | LTA | Bin | Fourier |
|---|---:|---:|---:|---:|
| Airport | 451 | 92.7% | 44.1% | 70.3% |
| Downtown | 455 | 7.3% | 9.9% | 39.3% |
| Sunset | 436 | 96.1% | 89.0% | 81.7% |
| Second Life | 1563 | 60.7% | 96.7% | 62.0% |
| Dining | 404 | 61.6% | 75.5% | 58.4% |
| Engineering | 940 | 95.3% | 51.3% | 27.4% |
| Medical | 207 | 72.0% | 79.2% | 40.1% |

We observe that LTA classifies a much higher number of Vagabonds than the other two methods in the Engineering area. Since the total appearance curve for this area is not amenable to partitioning the population into Vagabonds and Socials (Figure 4.2), the threshold selection method for LTA does not work well for this area.

We also conduct a pairwise comparison of the results of the three methods to determine to what extent they agree on device classifications. We use the fraction of users for which the methods make the same decision as the metric of similarity.

Table 4.3 compares the three methods. We observe that the overlaps are similar for LTA and Bin yet surprisingly different for Fourier, even though Bin and Fourier are both based on periodicity detection.

**Table 4.3:** Percentage of devices for which the classification methods agree.

| **Area** | **LTA & Fourier** | **LTA & Bin** | **Bin & Fourier** |
|---|---:|---:|---:|
| Airport | 71.8% | 51.4% | 53.4% |
| Downtown | 60.0% | 90.9% | 60.4% |
| Sunset | 81.4% | 91.5% | 78.4% |
| Second Life | 84.3% | 63.0% | 63.1% |
| Dining | 64.6% | 82.7% | 59.7% |
| Engineering | 23.2% | 56.0% | 55.5% |
| Medical | 49.8% | 85.0% | 52.2% |

For the remainder of the chapter, we use the Bin method to classify Socials and Vagabonds. Bin strikes a balance between the simplicity of LTA and the rigidity of Fourier. Although LTA is simple, the single dimensionality of appearance time is not flexible enough to capture essential differences in social behavior across the full range of areas. Fourier, however, requires Socials to appear according to a strict period and regimented schedule. Bin goes beyond LTA by incorporating appearance frequency and periodicity, but with a flexibility that better matches human behavior.

## 4.3 Contact properties

We know from previous work [CHC$^+$06, MMDA10] that contact characteristics are key in the effectiveness of opportunistic ad-hoc communication. We examine three different contact metrics: the *contact rate*, the *inter-contact time*, and the *contact duration*. We study these metrics for four different contact scenarios: *Social-meets-Socials* (SS), *Vagabond-meets-Socials* (VS), *Social-meets-Vagabonds* (SV) and *Vagabond-meets-Vagabonds* (VV). For example, the contact rate for VS is the rate at which a given vagabond device meets any social devices.

Our main observation is that Socials have significantly higher contact rates than Vagabonds, indicating that they have more opportunities for malware dissemination, while inter-contact times are heavier tailed for Vagabonds. This observation is in accordance with our expectations based on our definition of Vagabonds and Socials and provides a validation point for the classification method that we chose. However, we have also seen in Section 4.2 that Vagabonds considerably outnumber Socials in most regions. We later study how these two factors interact to affect malware dissemination in Section 4.4.

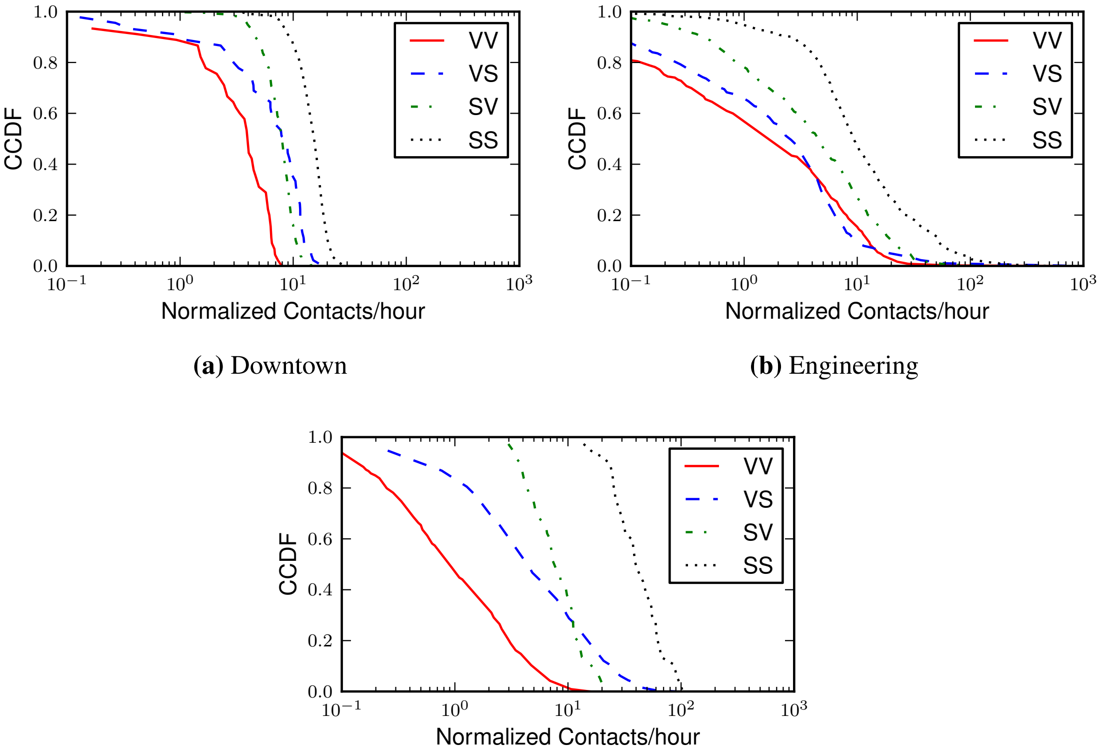
### 4.3.1 Contact rate

For each device, we compute the number of contacts per hour with other devices in the Social or Vagabond group. We normalize this metric to remove the bias introduced by the size of the target population. Figure 4.4 shows the CCDF of the normalized contact rates for representative areas of the three traces. We also chose these areas because they span the spectrum of Social and Vagabond combinations: Socials dominate Downtown SF, Vagabonds dominate Second Life, and they are balanced in Dartmouth Engineering. The results for the other areas are similar to these, and we omit the corresponding graphs for space considerations.

We observe in *all* areas that the SS contact rate is an order of magnitude higher than the VV contact rate, with the VS and SV contact rates somewhere in between. The distribution shape appears to be driven by the region characteristics and by the nature of the source device. The tail of the distribution is longer when the source device is a Vagabond (VS and VV contact rates), while SS and SV contact rate distributions decay faster and have short tails. This indicates that there are few Vagabonds that have higher contact rates than the rest of the vagabond population. This is possibly due to our method for selecting Socials and Vagabonds. Social devices exhibit quite homogeneous contact rates on the other hand.

### 4.3.2 Inter-contact time

The inter-contact time of a device is the time interval that starts with the end of a contact and ends with the beginning of the next contact, whatever the device encountered is. This quantity is very interesting as it characterizes the periods during which a device cannot forward any content to other devices. The inter-contact distribution has



**(a)** Downtown

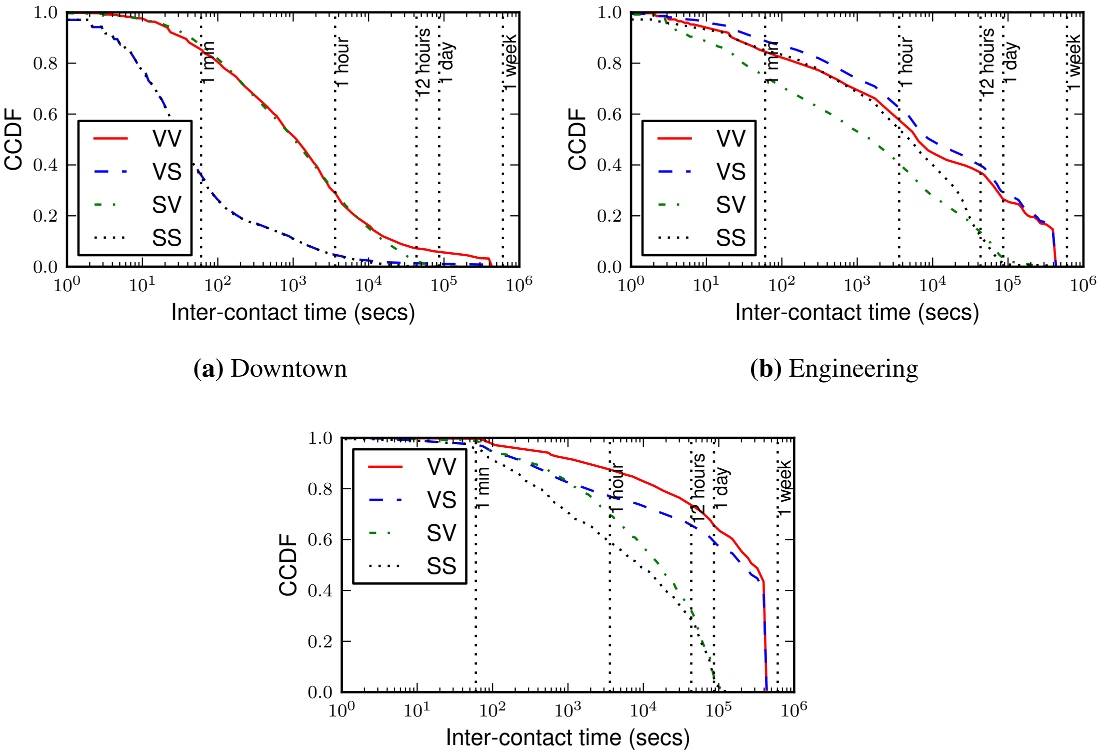
**(b)** Engineering

**(c)** Second Life

**Figure 4.4:** Contact rate distributions in three areas.

been shown to be heavily tailed [CHC$^+$06], which makes it impossible to estimate the delivery performance in such a network.

Figure 4.5 shows the CDF of the inter-contact time by social group of devices for the representative areas in each data set. We observe two different parts in each curve: the main body (roughly below 12 hours) and the tail of the distribution (above 12 hours). In the main body of the distribution, inter-contact is similar for Socials (respectively Vagabonds), independently of what type of device they encounter. This part of the distribution characterizes the mobility patterns that are specific to each area. The tails of the distribution though are always much longer when the device met is a Vagabond, independently of the nature of the source, which characterizes the vagabond devices and not the mobility in the area. This heavy-tailed inter-contact with Vagabonds will help us explain later why Vagabonds are not individually as effective at malicious content dissemination.



**(a)** Downtown

**(b)** Engineering

**(c)** Second Life

**Figure 4.5:** Inter-contact time distributions.

### 4.3.3 Contact duration

The amount of data that can be transmitted between two devices depends both on contact durations and on the communication technology (e.g., WiFi or Bluetooth). Therefore, contact duration is difficult to interpret and does not characterize the performance of communication in opportunistic ad-hoc networks. Contact duration is mostly a characteristic of the mobility in the area. As a consequence, we find that Socials and Vagabonds experience comparable contact characteristics and their distributions are very similar; as a result, we do not plot their distributions. In the Dartmouth data set, contacts last longer due to the stationary nature of the devices. Contacts are uniformly distributed between a couple of minutes and 3 hours. In San Francisco, the contact duration is defined by the road traffic condition in each area (with most of the cabs experiencing contacts between one second and one minute). In Second Life, avatar mobility is defined by social events or points of interest, which leads to the majority of contacts lasting between one minute and one hour.

## 4.4 Malware Dissemination

We now analyze the impact of each social group of devices on malware dissemination using trace driven simulations. We replay each trace multiple times using only Socials, only Vagabonds, or any device to propagate messages, while all devices can receive messages.

Our main observation is that, in areas in which Vagabonds outnumber Socials significantly, dissemination using Vagabonds outperforms dissemination using Socials, despite the lower contact rate experienced by Vagabonds. Further, we observe in most traces that there is a simple law by which we can predict which population is going to be more effective at propagating information.

### 4.4.1 Methodology

We simulate malicious message dissemination using flooding. Since the outcome depends on the start time of the simulation, we repeat the simulation by uniformly sampling many start times between the beginning of the selected week (Sunday midnight) and the middle of that week (Wednesday noon). At the start of each simulation only one device carries the message, and for each randomly chosen start time we simulate dissemination starting from each of the devices in the trace. Simulations last 2.5 days to ensure they all complete within the week-long trace. The number of simulations is determined by the standard deviation of the results of the completed simulations. For each point in time we calculate the average value and standard deviation of the number of devices receiving the message for all the completed simulations. We perform as many simulation runs as necessary so that each sampled point is within a 95% confidence of its expected value.

We also assume that message transfers are instantaneous. This simplification overestimates transmission opportunities, but it does not introduce a bias between Socials and Vagabonds as they exhibit similar contact durations characteristics.

We study two metrics that characterize malicious message dissemination in an area. The first characterizes the epidemic behavior of a population, while the second reports the optimal transmission delay in the network. *Contamination* is the number of devices that receive a given message as a function of time. Contamination reflects how effective a given population is at disseminating information in an area. In contrast, *shortest path* is the minimum time that is needed to reach a selected device. This metric characterizes the delay performance of propagation for each social group.

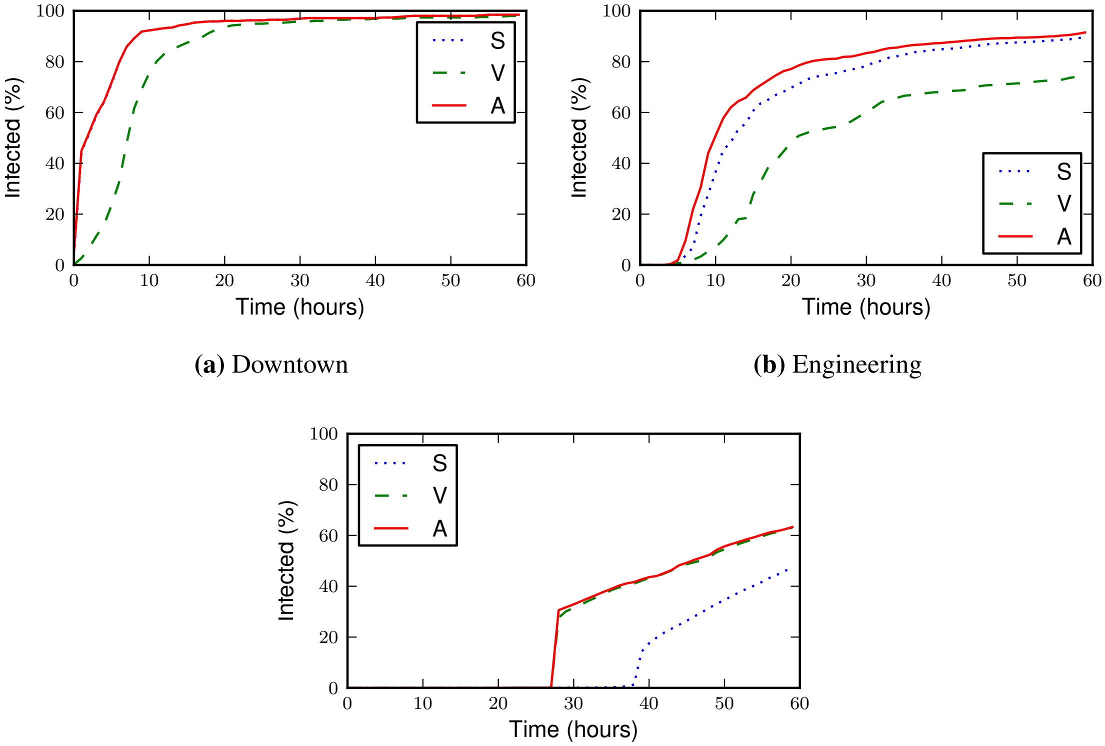
### 4.4.2 Contamination Evaluation

To understand the role that Socials and Vagabonds play in transmitting a message to the population of an area we first examine the number of devices that the message can reach relying only on Vagabonds or Socials. Note that we only account for message transmissions that take place through contacts that occur within the boundaries of the area. If devices make contact outside the area, we do not consider it to be a transmission opportunity since that situation does not reflect the contamination properties of a specific group of devices (the nature of a device being potentially different in each area).

Figure 4.6 shows the contamination result for the three different representative areas that we used previously. The curves represent the median across all simulations of the percentage of all devices reached.

The general observation is that Socials outperform Vagabonds in areas where they are the majority (SF Downtown) or of comparable population size (Dartmouth Engineering). However, in areas where Vagabonds largely dominate, they exhibit better contamination characteristics than Socials (Second Life). We also observed the same effect in all the other areas where Vagabonds form a clear majority (Dartmouth Dining and Medical, SF Sunset).

Individually, Socials contaminate more effectively than Vagabonds because they have a higher contact rate and more frequent contacts. In contrast, Vagabonds experience long periods of time without an opportunity to forward a message. However, we observe that large populations of Vagabonds can achieve the same contamination performance as Socials. Each Vagabond has a lower contact rate, but with many Vagabonds the total number of contacts is as high as what Socials would achieve with a smaller



**(a)** Downtown

**(b)** Engineering

**(c)** SL

**Figure 4.6:** Contamination within an area when using Vagabonds (V), Socials (S), or any device (A) to propagate messages.

number of devices.

To explore the relationship between the number of devices and social behavior further, we simulate malicious message dissemination while varying the population sizes of each group by taking random subsets. We decrease the number of Socials when the Social group performs better in an area, or similarly decrease the number of Vagabonds when they perform better, until we observe a similar contamination ratio for dissemination using each group. Table 4.4 reports these results. To have comparable contamination ratios, Vagabonds need to number two to six times more than Socials, depending on the area. Of course, these results are just one point in the parameter space balancing population sizes and social class—but they hint at the possibility of a deeper relationship. In the next Section, we formally present a model that develops a general "law" for this relationship.

**Table 4.4:** Vagabond and Social population sizes when contamination is comparable using either of the two groups.

| **Area** | **Better** | **Socials** | **Vagabonds** | **V / S** |
|---|---|---|---|---|
| Airport | S | 99 | 199 | 2.01 |
| Downtown | S | 22 | 45 | 2.04 |
| Sunset | V | 48 | 220 | 4.58 |
| Dining | V | 99 | 205 | 2.07 |
| Engineering | S | 229 | 482 | 2.10 |
| Medical | V | 43 | 140 | 3.26 |
| Second Life | V | 37 | 215 | 5.81 |

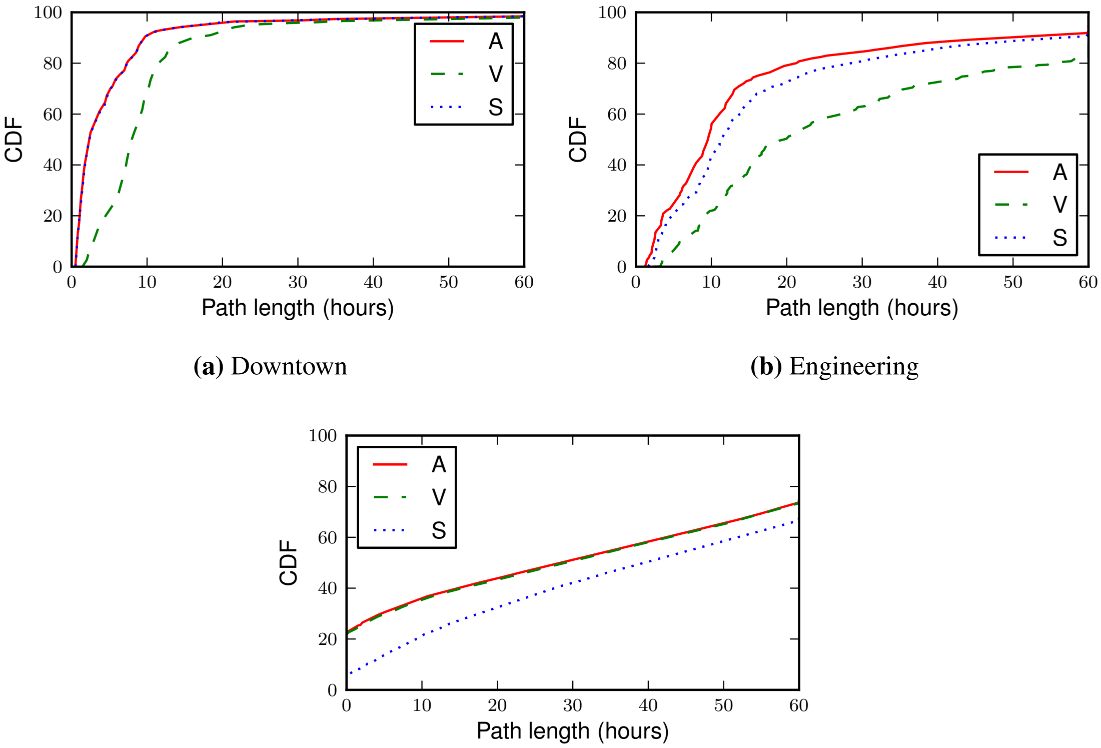
We learn here two major properties of communication in opportunistic ad-hoc networks. First, *the effectiveness of contamination is more a matter of contact "density" in an area than an issue of social behavior*. Second, *Vagabonds have an important role in dissemination of information and should not be ignored or removed when studying propagation in opportunistic networks*.

### 4.4.3 Shortest Path Evaluation

Lastly, we study a *shortest path* metric that measures the minimum time needed for a message to reach a specific device. We are interested in understanding if, because social devices visit an area frequently, messages to Socials will be delivered faster. We also want to understand if Vagabonds can act as "shortcuts" in message transmission. Therefore, we only study shortest path between any device and a Social.

For each device sending a message, we calculate the shortest path to any Social using only Socials, only Vagabonds, or any device. We calculate the median delay (i.e., the delay for 50% of the destinations) per message and we plot the distributions in Figure 4.7.

As with contamination, we observe than when Vagabonds significantly dominate in number (Second Life), they achieve best delays. In areas where Socials dominate or are comparable in number to Vagabonds, the shortest paths achieved by Socials are



**(a)** Downtown

**(b)** Engineering

**(c)** SL

**Figure 4.7:** Shortest paths within an area when using Vagabonds (V), Socials (S), or any device (A) to propagate messages.

very close to those obtained using all devices and Vagabonds do not seem to contribute significantly to reducing the delays.

As a result, in areas visited by many infrequent visitors, it is worthwhile sending a message to such visitors so that the message will arrive faster to someone that socializes within that area. In such a scenario, Vagabonds act as shortcuts in communication. Otherwise, it is better to prefer forwarding the message only to people that “socialize” within that area. The gain is two fold in this case. First, the delay in message delivery will be similar to using any device. Second, forwarding to fewer (but social) devices effectively reduces communication resource utilization.

## 4.5 Analysis

Section 4.4 indicates that the performance of malware dissemination depends both on the density of devices as well as their contact rate. As a result, even though Vagabonds have on average an order of magnitude fewer contact opportunities than Socials, they can achieve similar dissemination performance in areas with 3–4 times more Vagabonds than Socials.

The goal of this section is to formally characterize the relationship between the population size and the social behavior of users under which such phenomena occur. Our approach relies on a so-called "mean field" limit applied to epidemic dissemination.

### 4.5.1 Model Description

**Vagabonds and Socials**

We consider $N$ mobile users visiting an area $A$, partitioned into the two classes of Vagabonds and Socials. Let $N_v$ and $N_s$ be the number of Vagabonds and Socials, respectively. Users in each class enter and exit the area $A$ as follows. Time is slotted, and at each time slot a Vagabond enters $A$ with probability $\rho_v$, independently of previous slots and of other users. Similarly, a Social enters $A$ with a probability $\rho_s$. We call $\rho_v$ and $\rho_s$ the *occupancy rate* of Vagabonds and Socials, respectively, and we assume that $\rho_v \ll \rho_s$, i.e., Vagabonds spend less time in the area than Socials.

Note that the occupancy rate of each class captures the "social" behavior of the class, as it indicates whether its users frequent this area or not. The expected number of Vagabonds and Socials present in the area—i.e., the density of each class—is given by $\rho_v N_v$ and $\rho_s N_s$, respectively.

**Contacts between users and malware dissemination**

At each time slot, we select two users uniformly at random among all (unordered) pairs of the $N$ users in the system. If both of these users are within the area $A$ then a contact takes place between them. If at least one of them is outside $A$, then no contact takes place within this time slot. Note that, with $\rho_v \ll \rho_s$, the contact rate (average number of contacts per time slot) of a Social is higher than the contact rate of

**Table 4.5:** Summary of notation.

| | |
|---|---|
| $A$ | Visited area |
| $N$ | Total user population visiting $A$ |
| $N_v, N_s$ | Number of vagabond/social users |
| $r_v, r_s$ | Fraction of vagabond/social users |
| $\rho_v, \rho_s$ | Occupancy rate of vagabond/social users |
| $\lambda_{vv}$,$\lambda_{vs}$,$\lambda_{sv}$,$\lambda_{ss}$ | Transmission success probabilities |
| $I_v, I_s$ | Number of infected vagabond/social users |
| $S_v, S_s$ | Number of susceptible vagabond/social users |
| $i_v, i_s$ | Fraction of infected vagabond/social users |
| $s_v, s_s$ | Fraction of infected vagabond/social users |

a Vagabond, as the latter is far less likely to be inside $A$ at a given time slot. This is consistent with our empirical observations in Section 4.4.

Malware dissemination starts with an initial number of users (Vagabonds or Socials) carrying a message. Each time a user carrying the message contacts a user that does not, a message transfer occurs with a probability that depends on whether the two users are Vagabonds or Socials. As with the simulations in Section 4.4, we focus on the two cases where either Vagabonds or Socials (but not both) are message forwarders, while all devices can receive a message. In particular, denote by $\lambda_{vv}$, $\lambda_{vs}$, $\lambda_{sv}$, and $\lambda_{ss}$ the probabilities that transmissions succeed across and within classes; for example, $\lambda_{sv}$ is the probability that the message transfer succeeds when a Social contacts a Vagabond. We focus on the following two cases: (a) only vagabond users forward the message, i.e.,

$$\lambda_{vv} = \lambda_{vs} = 1, \text{ and } \lambda_{sv} = \lambda_{ss} = 0, \tag{4.1}$$

and (b) only social users forward the message, i.e.,

$$\lambda_{vv} = \lambda_{vs} = 0, \text{ and } \lambda_{sv} = \lambda_{ss} = 1. \tag{4.2}$$

### Main Result

Our analysis yields the following theorem, which quantifies when the “power of the crowd” dominates social behavior.

**Theorem 1.** *For large enough $N$, the epidemic dissemination using Vagabonds eventually dominates dissemination using Socials if and only if*

$$\frac{N_v}{N_s} > \left(\frac{\rho_s}{\rho_v}\right)^2. \tag{4.3}$$

Recall that Vagabonds occupy the area less frequently than Socials and are thus at a disadvantage w.r.t. epidemic dissemination. Theorem 1 implies that, when relative population sizes result in $N_v \gg N_s$, propagation using Vagabonds may outperform propagation using Socials. The necessary and sufficient condition is that the ratio of the two populations exceeds the square of the ratio of their occupancy rates. For instance, if Socials appear 10% of the time in the area, while Vagabonds appear only 5% of the time, Vagabonds will outperform Socials if their population is 4 times the population of Socials.

## 4.5.2 Proof of Theorem 1

### A fluid limit

Let $r_v = N_v/N$, $r_s = N_s/N$, be the corresponding fractions of the total population belonging to each class. We refer to users that carry the message as *infected* and users that do not as *susceptible*. We denote by $I_v$, $I_s$ the number of infected Vagabond and Socials, respectively, and by $i_v = I_v/N$, $i_s = I_s/N$ the corresponding fractions over all users. We also denote by $S_v$, $S_s$ the number of susceptible Vagabond and Socials, respectively, and by $s_v = S_v/N$, $s_s = S_s/N$ the corresponding fractions.

Under the assumptions of Section 4.5.1, the evolution of the vector $\vec{i}(t)$, $t \in \mathbb{N}$, representing the number of infected users in each class, is a stochastic process. Nonetheless, as $N$ tends to infinity, we can approximate the evolution of the system through a deterministic process, also known as a "fluid" or "mean field" limit. In particular, for large enough $N$, $\vec{i}(t)$ can be approximated with arbitrary accuracy through the solution of the following ordinary differential equation (ODE):

$$di_v/dt = \rho_v^2 i_v (r_v - i_v)\lambda_{vv} + \rho_v \rho_s i_s (r_v - i_v)\lambda_{sv} \tag{4.4a}$$

$$di_s/dt = \rho_s \rho_v i_v (r_s - i_s)\lambda_{vs} + \rho_s^2 i_s (r_s - i_s)\lambda_{ss} \tag{4.4b}$$

where the initial conditions $i_v(0)$ and $i_s(0)$ are set equal to the initial fractions of infected vagabond and social users. Note that the above ODE is essentially the classical susceptible-infected model (see, e.g., [NW07]) applied, in this case, to two infectious classes.

Formally, consider the following extension of the discrete time stochastic process $\vec{i} : \mathbb{N} \to [0,1]^2$ to a continuous time process $\vec{\underline{i}} : \mathbb{R}_+ \to [0,1]^2$. Define $\tau_k = \frac{k}{N}$, and, for all $k \in \mathbb{N}$,

$$\begin{aligned} &\vec{\underline{i}}(\tau_k) = \vec{i}(k), \text{ and} \\ &\vec{\underline{i}}(\tau_k + s) = \vec{i}(k) + s\frac{\vec{i}(k+1) - \vec{i}(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \frac{1}{N}. \end{aligned}$$

Observe that $\vec{\underline{i}}$ essentially evolves as $\vec{i}$, only it does so at a much faster timescale: for any integer $t$ we have that $\vec{i}(t) = \vec{\underline{i}}(t/N)$, so 1 time unit in $\vec{\underline{i}}$ corresponds to $N$ time slots in $\vec{i}$. Moreover, to define $\vec{\underline{i}}$ over all real numbers, $\vec{i}$ has been linearly interpolated: for $t \in \mathbb{N}$, and $\tau$ belonging to the real interval of the form $[\frac{t}{N}, \frac{t}{N} + \frac{1}{N}]$, the function $\vec{\underline{i}}(\tau)$ is a linear interpolation between the values $\vec{i}(t+1)$ and $\vec{i}(t)$.

Our main lemma states that the continuous version $\vec{\underline{i}}(\tau)$ of the fraction of infected users can be approximated with arbitrary accuracy through the solution of the ODE (4.4).

**Lemma 1.** *Let $\vec{\xi}(\tau)$, $\tau \in [0,\infty)$, be the solution of the ODE* (4.4) *with initial condition $\vec{\xi}(0) = \vec{i}(0)$. Then, for every $T \geq 0$, there exists $\varepsilon(N) = O(\frac{1}{N})$ such that*

$$\mathbf{P}\big(\sup_{0 \leq \tau \leq T} \|\xi(\tau) - \vec{\underline{i}}(\tau)\| \geq \varepsilon(N)\big) \leq \varepsilon(N).$$

*In other words,*

$$\lim_{N\to\infty} \sup_{0 \leq \tau \leq T} \|\xi(\tau) - \vec{\underline{i}}(\tau)\| = 0, \textit{ in probability}.$$

Intuitively, the above lemma implies that the trajectory of $\vec{i}(t)$, for $0 \leq t \leq T \cdot N$ (i.e., in an ever increasing interval), can be arbitrarily well approximated by the trajectory of the solution $\xi(\tau)$ of (4.4) in the interval $[0,T]$. For $N$ large enough, the probability that the stochastic process $\vec{i}(t)$ strays too far from the deterministic trajectory $\vec{\xi}(\tau)$ is arbitrarily small.

*of Lemma 1.* Under the contact assumptions presented in Section 4.5.1, we have that the probability that $I_v$, the number of infected Vagabonds, increases by one at time $t+1$ is given by

$$f_v(\vec{i}(t)) = \rho_v^2 i_v(t) s_v(t) \lambda_{vv} + \rho_s \rho_v i_s(t) s_v(t) \lambda_{sv}$$

To see this equation, observe that the probability that a contact of an infected vagabond user inside area $A$ is selected is $\rho i_v(t)$, while the probability that an uninfected vagabond user inside area $A$ is selected is $\rho s_v(t)$, yielding the product in the first term of the above sum; the second term can also be derived using the same intuition. Similarly, the probability that the number of infected Socials increases by one can also be shown to be

$$f_s(\vec{i}(t)) = \rho_v \rho_s i_v(t) s_s(t) \lambda_{vs} + \rho_s^2 i_s(t) s_v(t) \lambda_{vs}.$$

From the above, we get that:

$$\mathbb{E}[i_v(t+1) - i_v(t) \mid i_v(t), i_s(t)] = \frac{1}{N} f_v(\vec{i}(t))$$
$$\mathbb{E}[i_s(t+1) - i_s(t) \mid i_v(t), i_s(t)] = \frac{1}{N} f_s(\vec{i}(t))$$

where $f_v, f_s$ are continuously differentiable. Moreover, observe that the change of either $I_V$ or $I_S$ in each time slot is at most one. As a result, the assumptions H1–H5 of Theorem 1 in Benaïm and Le Boudec [BB08] are satisfied; the lemma follows directly from the above theorem. □

**Solution of the ODE** (4.4)

The following lemma determines the evolution of $\vec{i}(t)$, as given by (4.4), under a single infectious class. population when only one class is infectious.

**Lemma 2.** *The ODE*

$$dx/dt = \alpha(A-x)x \tag{4.5a}$$

$$dy/dt = \beta(B-y)x \tag{4.5b}$$

*with initial conditions $x_0$, $y_0$, has the solution*

$$x(t) = A - (A-x_0)A/\left(x_0 e^{\alpha A t} + (A-x_0)\right) \tag{4.6a}$$

$$y(t) = B - (B-y_0)\left[A/\left(x_0 e^{\alpha A t} + (A-x_0)\right)\right]^{\beta} \tag{4.6b}$$

*Proof.* From (4.5a), we have that $\frac{dx}{(A-x)x} = \alpha dt$. Integrating both parts and considering the initial condition $x(0) = x_0$ yields (4.6a). Note also that, from (4.5a) $x = \dot{x}/(A-X)$ so

$$\int x dt = -\log(A-x) + c \tag{4.7}$$

where $c$ a constant. On the other hand, from (4.5b), we have that $\frac{dy}{(B-y)} = \beta x dt$. Integrating both parts of this equation and using (4.7) yields $-\log(B-y) = -\beta \log(A-x) + c'$ which, in turn, along with (4.6a) and the initial condition $y(0) = y_0$ yields (4.6b). □

Lemma 2 can be used to describe the evolution of the infected population when only one of the two classes is infectious. If, for example, only Vagabonds are infectious then conditions (4.1) will hold; Lemma 2 then applies with $\alpha = \rho_v^2$, $\beta = \rho_v\rho_s$, $A = r_v$, and $B = r_s$. We thus obtain that $\vec{i}^{vo}(t)$ is given by:

$$i_v^{vo}(t) = r_v - (r_v - i_v^{vo}(0)) \frac{r_v}{i_v^{vo}(0)e^{\rho_v^2 r_v t} + r_v - i_v^{vo}(0)} \tag{4.8a}$$

$$i_s^{vo}(t) = r_s - (r_s - i_s^{vo}(0)) [\frac{r_v}{i_v^{vo}(0)e^{\rho_v^2 r_v t} + r_v - i_v(0)^{vo}}]^{\rho_v\rho_s} \tag{4.8b}$$

Similarly, if only social users are infectious, then conditions (4.2) will hold; by taking $\alpha = \rho_s^2$, $\beta = \rho_v\rho_s$, $A = r_v$, $B = r_s$, we obtain that $\vec{i}^{so}(t)$ is given by:

$$i_s^{so}(t) = r_s - (r_s - i_s^{so}(0)) \frac{r_s}{i_s^{so}(0)e^{\rho_s^2 r_s t} + r_s - i_s^{so}(0)} \tag{4.9a}$$

$$i_v^{so}(t) = r_v - (r_v - i_v^{so}(0)) [\frac{r_s}{i_s^{so}(0)e^{\rho_s^2 r_s t} + r_s - i_s^{so}(0)}]^{\rho_v\rho_s} \tag{4.9b}$$

Using the above, we establish that the condition of Theorem 1 implies the domination of propagation through Vagabonds.

**Lemma 3.** *Let $i^{vo}$ and $i^{so}$ be the fractions of infected users under ODE* (4.4) *when either* (4.1) *or* (4.2) *hold, respectively. If $\rho_v^2 r_v > \rho_s^2 r_s$, then* $\lim_{t\to\infty}(1 - i^{vo}(t))/(1 - i^{so}(t)) = 0$.

*Proof.* By (4.8), if only the Vagabonds are infectious then $1 - i^{vo}(t) = \Theta(e^{-\rho_v^3 \rho_s r_v t})$. Similarly by (4.9), if only the social users are infectious then $1 - i^{so}(t) = \Theta(e^{-\rho_s^3 \rho_v r_s t})$. Hence, for large $t$, the dissemination if Vagabonds are used will dominate if and only if $\rho_v^2 \rho_s r_v > \rho_s^3 \rho_v r_s$. □

Theorem 1 therefore follows directly from Lemmas 3 and 1. To summarize, it implies that, if $\rho_v^2 r_v > \rho_s^2 r_s$, the propagation using Vagabonds eventually dominates the propagation using social users, in spite of the fact that Vagabonds show up in the area much less frequently than Socials.

### 4.5.3 Numerical Validation

Figure 4.8(a) illustrates the performance of epidemic propagation under our model, evaluated through the ODE (4.4). We consider population ratios $N_v/N_s$ ranging between 0.1–10 and occupancy rates $\rho_s$, $\rho_v$ ranging between 1–10%. Circles correspond to cases for which propagation using Socials infects 97% of the population faster, and crosses are cases when propagation using Vagabonds is faster. The dashed line corresponds to a balance in propagation speeds between Vagabonds and Socials, as predicted by the inequality in Theorem 1.

Note that Theorem 1 is asymptotic: it states that when $N_v\rho_v^2 > N_s\rho_s^2$, Vagabonds will *eventually* dominate Socials. Figure 4.8(a) shows that correctly predicts which class reaches the 97% contamination threshold in most cases. The cases for which the theorem does not correctly predict the outcome are due to insufficient time for the asymptotic behavior to manifest; indeed, we repeated these evaluations with higher thresholds and observed a decrease in misclassified points.

Recalling the simulations in Section 4.4, none reached more than 95% of the total population, so it is difficult to compare the analytic results in Figure 4.8(a) with our simulation results. Instead, Figure 4.8(b) shows the relative propagation performance of Vagabonds and Socials after 60 hours of message propagation. Circles correspond to cases where, after 60 hours, the simulated propagation using Socials infected more users than the propagation using Vagabonds, while crosses correspond to the converse. To exclude simulations not in the asymptotic regime, we show only the cases where either simulation reached more than 60% of the total population. Although many of these points are far from the asymptotic propagation behavior, Theorem 1 correctly predicts the outcome in most cases.

In summary, we proposed a model incorporating the population sizes of Social and Vagabond devices, as well as their social behavior. We have identified a law de-

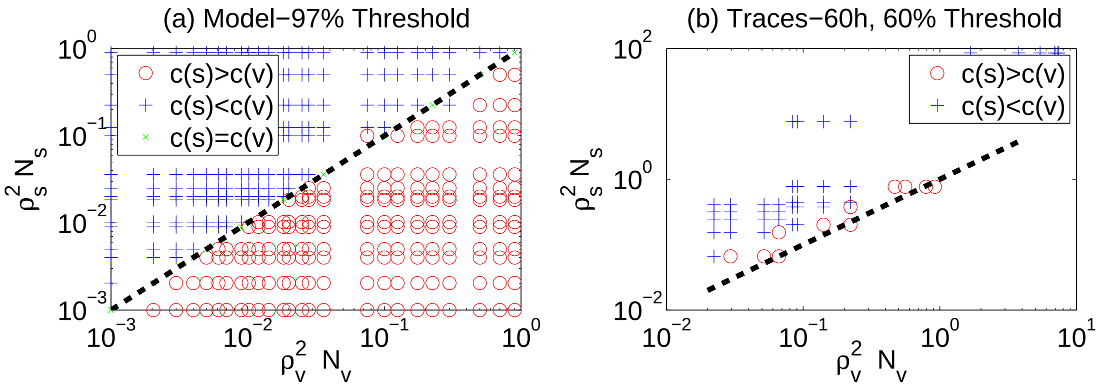


**Figure 4.8:** Validation of Theorem 1: (a) relative performance of epidemic propagation using Vagabonds and Socials under our model; (b) relative propagation performance using Vagabonds and Socials from the Dartmouth, SF, and SL traces. $r_s = N_s/N$ while $\rho_v$ and $\rho_s$ are the occupancy rates. The dashed lines indicate the threshold above which, according to Theorem 1, Vagabonds outperform Socials.

termined by these two parameters that governs the asymptotic efficiency of epidemic dissemination. Though our focus was on asymptotic efficiency, our ODE approach in general applies to more complicated interactions between users, including, e.g., transmissions that fail with class-dependent probabilities or re-infections introduced after a received message expires.

## 4.6 Summary

In this chapter we improved our understanding of malware dissemination in opportunistic mobile ad-hoc networks. By separating users into two behavioral classes, we find that, although Socials form an active population subset, most areas are dominated by Vagabonds in terms of population size. Vagabonds, often excluded as unimportant, can often play a central role in opportunistic networks. As a result, tracing efforts should strive to capture the presence of Vagabonds, and analyses of protocols and applications should not discount them.

This work is just a first step in studying the impact of social behavior of users on malware dissemination. A number of interesting directions naturally follow, including studying the characteristics of inter-area message propagation, the dynamics of user social behavior (e.g., Vagabonds becoming Socials in other areas), and the interactions between Vagabonds and Socials in supporting malware dissemination.

Chapter 4, in part, is a reprint of the material as it appears in the “Proceedings of the IEEE Infocom Conference 2011” with the title “Dissemination in Opportunistic Mobile Ad-hoc Networks: the Power of the Crowd” by Gjergji Zyba, Geoffrey M. Voelker, Stratis Ioannidis, Christophe Diot. The dissertation author was the primary investigator and author of this paper.

# Chapter 5

# Inferring Human Contacts Using Scanners

Proximity malware propagates over short-range radio connections (e.g., Bluetooth [SCM$^+$06]), as we saw on Chapter 3. Understanding the dynamics of propagation and assessing the effectiveness of countermeasures in outbreaks of self-replicating computer malware relies on understanding two factors: the mechanism by which the malware infects a susceptible host, and the patterns of contacts between hosts. For network-borne malware these contacts are practically instantaneous and enabled by network topology rather than spatial relationships. For proximity-borne malware (which applies only to mobile devices) the patterns of contact between people carrying the devices are critical in developing an understanding of the propagation dynamics. Chapter 4 explored how human social behavior impacts these contacts. However, many studies, including our study on Chapter 4, infer real world contacts from observations of devices which are either mobile or static (e.g., WiFi Access Points, Bluetooth scanners). In this chapter we explore the benefits and limitations of this methodology.

When attempting to understand and model proximity-based propagation the availability of relevant and generalizable empirical data is limited. In this chapter, we consider the approach of deploying scanners which use the same radio technology as devices carried by users. These scanners connect to users' devices when they pass within range and store information about the detected devices. Devices which are detected simultaneously by a given scanner are considered to be spatially co-located (corresponding to observed "encounters" within the scanner's radio range). The main benefit of such an approach is that once the scanners are deployed, large amounts of data can be gathered easily and at low cost, allowing longitudinal comparisons of encounter patterns. However, there are also some drawbacks when such scanner data is used as a basis for inferring proximity-based malware propagation dynamics: *(i)* Scanner deployments in the real world tend to be of relatively low density, typically covering a small fraction of an area under consideration, such as a campus or a part of a city. Hence, within this already limited area, the majority of encounters between devices will take place out of range of the scanners. *(ii)* Moreover, as we limit the area of consideration, we would expect the frequency with which particular devices appear at any scanner to decrease, artificially lengthening device inter-contact times.

Despite these obvious limitations, if scanned data is used carefully (i.e., accounting for the effects of missed encounters) it would still appear to be a good source of empirically-derived data on human encounters. Superficially, the encounters which *are* captured should consist of a subset of the actual device encounters taking place in the area under study at a particular time. In fact, we found that the process of inferring copresence encounters between pairs of devices based on empirical evidence of simultaneous sightings by third-party scanners leads to the introduction of errors. We investigated the extent to which errors are introduced, and made the following contributions.

- We derived analytical results on errors introduced in scanner-based measurements for a simplified case where all scanners and devices are static, and where radio propagation details are omitted. We examined the differences between device copresence as inferred by the scanners and actual copresence between the devices, and classified the discrepancies.

- Based on this classification, we then derived the probabilities with which each type of discrepancy will occur. Using simulation we validated our analytical finding that approximately 41% of copresence encounters inferred by scanners do not correspond to actual device copresence.

- Also using simulation, we demonstrated the extent and impact of errors when device mobility is included. As a concrete application, we studied the effect on proximity-based malware propagation, an example of flooding-based data dissemination which depends heavily on the patterns of device encounters. We found that, in addition to the expected cases of missed and spuriously inferred encounters, the set of encounters inferred from scanners differs from the actual encounters simulated in the model in terms of duration distribution and probability of encountering previously unmet devices. While the magnitude of these errors increases when simulated mobility is more diffusive, in all the cases we considered malware propagation models showed slower propagation using scanned encounters compared to actual encounters for devices with the same mobility characteristics.

## 5.1 From device contacts to inferred encounters

The data captured by scanners is not necessarily an accurate representation of the real contacts taking place between mobile devices—even if we consider only the subset of real contacts taking place within the scanner's range. Scanners infer copresence encounters when a device pair is simultaneously sighted at the same scanner. We assume throughout that the scanner has the same radio range as the devices. Simple geometry indicates that a scanner, if capable of the same radio range as the devices moving around it, will be able to make contact with pairs of devices which are simultaneously within range of the scanner, but not within range of each other. This effect, which we term "bridging", leads to the incorrect inference of encounters between devices which did not actually meet (see Figure 5.2).

### 5.1.1 Static Analysis

To begin to understand the relationship between scanned encounters and actual contacts between mobile devices, we first consider a simplified case: a single time instance in which all devices and scanners are static. We derive simple expressions for the expected number of different encounter types as seen by an array of fixed scanners.

We consider *n* devices, each equipped with a short-range radio. We assume that this radio behaves ideally, producing a disc of constant signal strength with radius *r*. The devices are uniformly distributed over a rectangular area of size $a \times b$, and are observed by $m$ scanners using the same radio technology and placed in the same area. We assume that the coverage areas of scanners do not overlap.

Let $\mathbf{X}_i$ $(i = 1, \ldots, n)$ denote the position of each device. We assume that these 2-dimensional random vectors $\mathbf{X}_i$ are independent and identically uniformly distributed (*iid*) over the rectangle $[0,a] \times [0,b]$. Using a simplified "perfect disc" radio propagation assumption, we say that two devices $i$ and $j$ are in contact if they are within radio range, that is $\|\mathbf{X}_i - \mathbf{X}_j\| \leq r$. With a slight abuse of notation, we write $r(\mathbf{X}_i, \mathbf{X}_j)$ as a shorthand for this relation, and $\overline{r}(\mathbf{X}_i, \mathbf{X}_j)$ as a shorthand for its negation (i.e., $\|\mathbf{X}_i - \mathbf{X}_j\| > r$).

Similarly let $\mathbf{y}_k$ $(k = 1, \ldots, m)$ denote the scanner positions. Then, using our shorthand notation, the event that device $i$ is in range of the $k^{th}$ scanner can be expressed as $r(\mathbf{X}_i, \mathbf{y}_k)$.

In our model, a scanner $k$ registers an *inferred contact* between two devices, $i$ and $j$, if both devices are simultaneously within the scanner's range; or, more formally:

$$r(\mathbf{X}_i, \mathbf{y}_k) \wedge r(\mathbf{X}_j, \mathbf{y}_k).$$

Note that an inferred contact need not correspond to an actual device contact, since two devices may both be in range of the same scanner without being in range of one another (the "bridging effect"). On the other hand, not all actual device contacts will be inferred by a scanner, since either one or both of the devices involved in a contact with each other may be outside scanner range. The following analysis classifies all possible relationships between device contacts and the contacts inferred by scanners.

### Types of Contacts Considered

Not all device contacts can be inferred by scanners, and not all contacts which the scanners do observe correspond to actual contacts between devices. We set out to derive expressions for the expected number of:

- real contacts between devices,
- contacts inferred by scanners (consisting of):
  - correctly inferred contacts ("inferred real"),

  - incorrectly inferred contacts (“inferred fake”),

- real device contacts missed by scanners, because
  - one device is outside coverage (“missed one”),
  - both devices are outside coverage (“missed two”).

Figure 5.1 shows a diagram relating these contact types. We assume a sufficiently sparse scanner arrangement to preclude missed contacts where both devices are in scanner range, but the two devices are in the range of two *different* scanners.

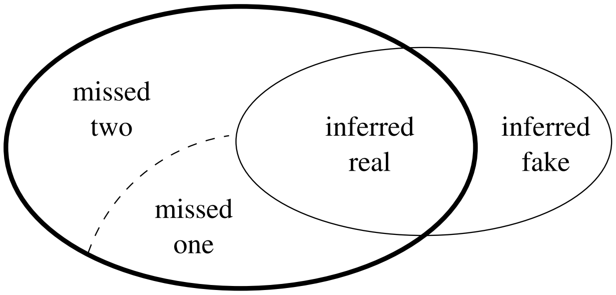


**Figure 5.1:** Contact types: real (thick ellipse) and inferred (thin ellipse) contacts

### Real Contacts

Intuitively, the expected number of device contacts (without regard to which of these are inferred or missed by the scanners) for *n iid* devices should equal the number of possible device pairs, $\binom{n}{2}$, times the probability of contact between any two devices. In the following derivations $\mathbb{P}$ denotes probabilities. Let us denote the ratio of a coverage area to the total observation area by

$$p \doteq \frac{r^2\pi}{ab}. \tag{5.1}$$

Now, one can write the expected number of real contacts as

$$c_{\text{real}} = \binom{n}{2} p(1-\delta) \tag{5.2}$$

where $\delta$ is an error term accounting for border effects. In the model, the border effects can be removed, e.g., by having edge wrap-around (effectively forming a torus), in which case the error term can be omitted.

### Observed Contacts

For a single scanner at position $\mathbf{y} \in [r, a-r] \times [r, b-r]$, we have

$$\mathbb{P}[r(\mathbf{X}_i, \mathbf{y})] = p, \qquad \forall i.$$

Thus, the expected number of inferred contacts for this scanner equals $\binom{n}{2}p^2$; i.e., the number of device pairs times the probability that both devices "independently" fall inside the scanner's coverage area. For $m$ scanners, whose coverage areas do not overlap and lie completely inside the measurement area, the above probability for a single scanner is simply multiplied by $m$ to yield the expected number of inferred contacts.

$$c_{\text{inferred}} = m\binom{n}{2}p^2 \approx mp\, c_{\text{real}} \tag{5.3}$$

The result may appear intuitively satisfying, as it suggests that the scanners capture the fraction of real contacts corresponding to their combined coverage area. Unfortunately, this intuition is somewhat misleading, since a sizable portion of these inferred contacts are in fact "fake" and result from bridging, as we show next.

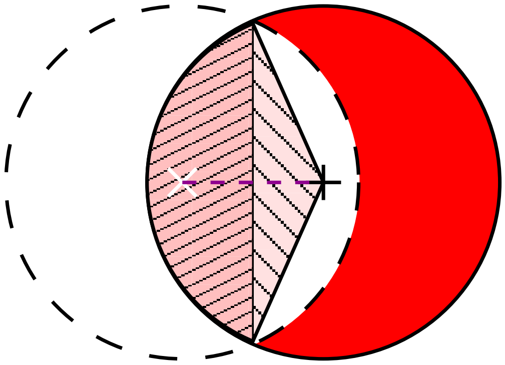


**Figure 5.2:** Bridging probability. The position and coverage area of the scanner are shown by a $+$ and the solid circle, while those of the first device by an $\times$ and a dashed circle. If the position of the second device falls inside the dark-shaded area to the right, the scanner will infer a contact while none actually occurs.

The bridging probability $\beta$ is the conditional probability that two devices both inside the same scanner's coverage area are not within range of one another. Pictorially speaking, this situation corresponds to the average fraction of the darker half-moon-shaped area to the right in Figure 5.2. While we omit the full derivation for brevity, we have

$$\beta \doteq \mathbb{P}[\bar{r}(\mathbf{X}_1,\mathbf{X}_2)\,|\,r(\mathbf{X}_1,\mathbf{y}_1),r(\mathbf{X}_2,\mathbf{y}_1)] = \frac{3\sqrt{3}}{4\pi} \tag{5.4}$$

In other words, over 41% of inferred contacts are fake, introduced by bridging; and, consequently, only about 59% of inferred contacts correspond to real device contacts. The resulting formulas for the expected number of real and fake inferred contacts can be written as follows.

$$c_{\text{inferred,real}} = m\binom{n}{2}p^2(1-\beta) \qquad \approx 0.5865\; c_{\text{inferred}} \tag{5.5}$$

$$c_{\text{inferred,fake}} = m\binom{n}{2}p^2\beta \qquad \approx 0.4135\; c_{\text{inferred}} \tag{5.6}$$

### Missed Contacts

As noted earlier, due to our assumptions on scanner placement, we consider only two types of missed contacts. In the first, one of the devices is inside scanner range while the other one is outside; and in the second type, both devices are outside scanner range.

Another careful look at Figure 5.2 reveals that the conditional probability of a device outside a given scanner's range being in range of another device inside the same scanner's range also equals the bridging probability $\beta$ (as this case corresponds to the unshaded half-moon-shaped area on the right in the figure). Given that, viewed as an ordered pair, either of the two devices in this type of missed contact could be inside or outside scanner range, the following formula obtains:

$$c_{\text{missed,one}} = 2m\binom{n}{2}p^2\beta = 2\; c_{\text{inferred,fake}} \approx 0.827\; c_{\text{inferred}} \tag{5.7}$$

where one factor of $p$ corresponds to $r(\mathbf{X}_1,\mathbf{y}_1)$ (i.e., the probability that the "first" device is in scanner range) and the other to $r(\mathbf{X}_1,\mathbf{X}_2)$ (i.e., that the two devices are within range of one another).

When both devices are outside scanner range, the analysis naturally splits into two sub-cases: one that accounts for border effects around the edges of the observation area, and another that accounts for similar effects near each scanner, when the scanner's and the devices' coverage areas intersect, as illustrated in Figure 5.3.

The geometry of the dark shaded area in Figure 5.3 is essentially the same as the geometry of the dark shaded area in Figure 5.2, with the notable exception that

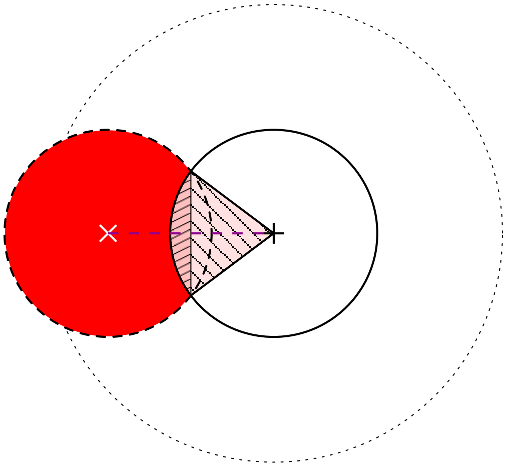


**Figure 5.3:** Border effect near a scanner. If the first device is within distance $2r$ of a scanner (dotted circle), the position of the second device must fall inside the dark-shaded area to the left in order to guarantee that both devices end up outside scanner range.

the distance between the scanner's and the device's position varies between $r$ and $2r$ (compared to 0 and $r$ for the bridging probability). Consequently, $\lambda_k$, the corresponding conditional probability for scanner $\mathbf{y}_k$, can be obtained in the same fashion, giving:

$$\begin{aligned}\lambda_k &\doteq \mathbb{P}\left[\bar{r}(\mathbf{X}_2,\mathbf{y}_k)\,|\,r(\mathbf{X}_1,\mathbf{X}_2),2r(\mathbf{X}_1,\mathbf{y}_k),\bar{r}(\mathbf{X}_1,\mathbf{y}_k)\right] \\ &= 1-\frac{\sqrt{3}}{4\pi} = 1-\frac{\beta}{3} \qquad (5.8)\end{aligned}$$

Using equation (5.8) and subject to the constraints that:

a) the distance between any two scanners is at least $4r$, which avoids simultaneous interactions with multiple scanners, and

b) the distance between any scanner and the edge of the observation area is at least $3r$, excluding interactions between scanners and edge effects,

the expected number of missed contacts due to both devices being outside scanner coverage can be derived as follows.

$$\begin{aligned}
c_{\text{missed,two}} =& \binom{n}{2}\mathbb{P}\left[r(\mathbf{X}_1,\mathbf{X}_2), \bigwedge_l \bar{r}(\mathbf{X}_2,\mathbf{y}_l), \bigwedge_l \bar{r}(\mathbf{X}_1,\mathbf{y}_l)\right] \\
=& \binom{n}{2}\left(\mathbb{P}\left[r(\mathbf{X}_1,\mathbf{X}_2) \middle| \bigwedge_l \overline{2r}(\mathbf{X}_1,\mathbf{y}_l)\right](1-\delta-4mp)\right. \\
& \left. + \sum_k \lambda_k p(4p-p)\right) \\
=& \binom{n}{2}\left(p(1-\delta-4mp)+m(4p-p)p(1-\frac{\beta}{3})\right) \\
=& \binom{n}{2}p\left(1-\delta-mp(1+\beta)\right) && (5.9) \\
=& c_{\text{real}} - \left(c_{\text{inferred,real}} + c_{\text{missed,one}}\right) && (5.10)
\end{aligned}$$

### 5.1.2 Validation

To validate our analysis, we performed a simple simulation. Using the same constraints as above on scanner arrangement, we randomly placed 5,000 devices and 144 scanners, both with a range $r = 10m$, within a simulated area of size $500m \times 500m$. We then compared the encounters recorded directly by the devices with the encounters inferred by the scanners over 100 simulation runs.

In our simulation an average of 41.347% (SD=1.29%) of encounters inferred by the scanners were "fake", that is, they did not correspond to pairs of devices which were in range of each other. We further found that the scanners was 82.91% (SD=4.40%). the mean number of pairwise encounters missed by scanners because only one device was in range, divided by the total number of encounters which were inferred by scanners, was 0.829 (SD=0.044). These values compare to the expectations of 41.35% (Equation 5.6) and 0.827 (Equation 5.7) which were derived in our previous analysis.

## 5.2 Scanner errors in mobile data

In Figure 5.1 we defined the four possible classifications of contact types in a static scenario: *inferred-real*, *inferred-fake*, *missed-one*, *missed-two*. These contact

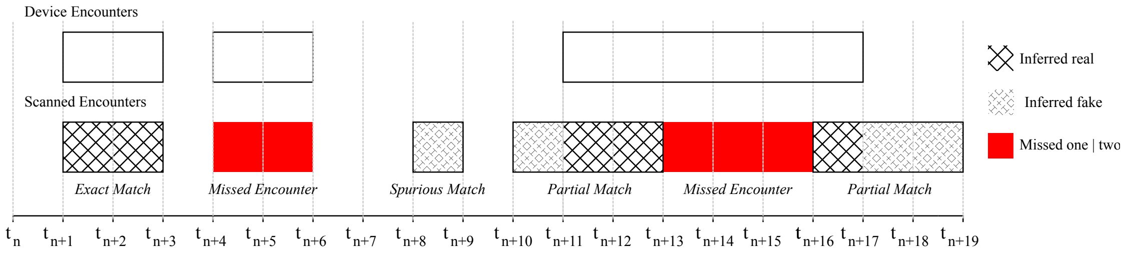

Device Encounters
Scanned Encounters
Exact Match
Missed Encounter
Spurious Match
Partial Match
Missed Encounter
Partial Match
Inferred real
Inferred fake
Missed one | two
$t_n$ $t_{n+1}$ $t_{n+2}$ $t_{n+3}$ $t_{n+4}$ $t_{n+5}$ $t_{n+6}$ $t_{n+7}$ $t_{n+8}$ $t_{n+9}$ $t_{n+10}$ $t_{n+11}$ $t_{n+12}$ $t_{n+13}$ $t_{n+14}$ $t_{n+15}$ $t_{n+16}$ $t_{n+17}$ $t_{n+18}$ $t_{n+19}$

**Figure 5.4:** Relating device contacts to scanned encounters

types describe all the ways in which a scanner might (or might not) register an encounter between two devices at a particular moment in time.

When mobility and time are introduced, the way in which a scanner infers (or misses) an encounter can be thought of as a sequence of successive static contacts over a period of time. The relationship between the device encounter and what is inferred (or not) by the scanner falls into four possible classifications, each characterized by which of the four contact types makes up the sequence. We showed the relationship between device encounters and scanned encounters, and their corresponding contact types, in Figure 5.4. The top time-series shows the ground truth of the encounters of two devices, and the bottom time-series shows the different ways a scanner may infer (or miss) device encounters. The shading on each scanned encounter relates to the contact types from which it is composed over time. While we separately considered the distinct probabilities of *missed-one* and *missed-two* in our static analysis, the distinction is not useful from here on; in both cases a device encounter goes undetected by all scanners.

- **Exact matches.** Encounters between two devices that take place entirely within scanner range. Devices are only within range of a scanner for the period of time that they are within range of each other. These encounters are composed entirely of *inferred-real* contacts, as in the encounter in Figure 5.4 beginning at time $t_{n+1}$.

- **Missed encounters.** Encounters between two devices that take place beyond the range of scanners are composed of *missed-one* or *missed-two* contacts. In Figure 5.4 this situation is shown by an unbordered solid area at time $t_{n+4}$.

- **Spurious matches.** Encounters between two devices when the devices are each within range of a scanner, yet are never actually in range of each other. These encounters are composed entirely of successive *inferred-fake* contacts, as in the encounter in Figure 5.4 at time $t_{n+8}$.

- **Partial matches.** Encounters between pairs of devices which are composed of more than one of the four types of contact (other than a mixture of *missed-one* and *missed-two*). Compared to the actual encounter the scanner may infer one or more longer or shorter encounters which partially overlap in time with the actual encounter. In Figure 5.4 two such encounters are shown beginning at times $t_{n+10}$ and $t_{n+16}$.

The extent to which the encounters inferred (or missed) by scanners fall into each of these categories is critical when considering the degree of error which is likely to appear in malware propagation models based on them.

Consider an ideal scanner, which was somehow able to accurately infer all device encounters which passed within its range. For this scanner, all of its inferred encounters would fall into one of the first two categories above. As such, they would be composed entirely of *inferred real* and *missed one* or *missed two* contacts. By definition, the encounters inferred by this scanner would be a subset of the total set of device encounters which took place in the area under study. While those encounters which never passed in range of the scanner would be missed, if we assume that devices are dispersed uniformly around the area then we would expect the selection of encounters which the scanner did infer to be unbiased. As such, a deployment of a number of these ideal scanners would together collect a subset of total device encounters (once de-duplicated to account for encounters which pass through multiple scanners). We would expect the relative numbers of encounters inferred to reflect the ratio of scanner coverage area to the total area under study. If we compared the property distributions (e.g., encounter duration) of the scanner and device encounters, we would expect to see no difference. As a consequence, once we adjusted for the lower encounter rates in the scanner data, we would expect to see identical dynamics of propagation between the two sources of encounter data.

In practice, we can easily identify ways in which scanners deviate from ideal behavior. In our previous analysis, we have shown that bridging leads scanners to incorrectly infer copresence in around 41% of device pair sightings. These incorrect inferences will give rise to *partial matches* and *spurious matches* appearing in the scanner data, which differ from, or do not actually appear in, the device encounters. In addition, when we consider moving devices over time, even in the absence of bridging, scanners can report encounters which differ from actual device encounters. For example, a pair of devices may remain in range of each other while moving on equal vectors. They may pass in and out of the ranges of a number of scanners, which will report numerous shorter encounters between the pair, instead of one continuous meeting. In this particular example, the scanner data will contain a number of *partial match* encounters. If we again consider the distribution of encounter duration in the scanner encounters, a disparity will clearly be introduced compared to the device encounters.

Having shown the circumstances in which the use of scanners to infer copresence can introduce errors, the remainder of the chapter investigates how often these errors occur, and the extent to which they lead to inaccurate estimates of proximity borne malware propagation as one concrete application.

### 5.2.1 Methodology

Compared to our analytical solution for the static case, a similar analytic model for the dynamic case of devices that are mobile over time is substantially more complex. On the other hand, obtaining empirical data with which to compare the incidence of errors in scanned encounters is also difficult. To understand the extent of errors introduced by the use of scanners in the case of mobile devices, we require data on both the real encounters (as detected by devices themselves) and the encounters inferred from scanners for the same set of devices at the same time. Most datasets consist of either high volume scanned data or relatively low volume GPS trace data, but not both.

Lacking empirical data, we instead used a mobility simulator to produce complete traces of mobile devices moving within a simulated two-dimensional space. To obtain a baseline for the actual encounters between devices, we processed the mobility traces generated by the simulator to identify the encounters between pairs of devices over time. Using the same definition as in the static case above, we say that a pair of devices $i$ and $j$ with positions given by two-dimensional vectors $\mathbf{X}_i$ and $\mathbf{X}_j$ are within radio range while $\|\mathbf{X}_i - \mathbf{X}_j\| \leq r$. As the devices move over time, we say that they are in a pairwise encounter for any contiguous time period during which they remain in radio range, i.e., for an encounter between times $t_m$ and $t_n$ (where $m < n$):

$$\|\mathbf{X}_i(t_s) - \mathbf{X}_j(t_s)\| \leq r \qquad \forall s : m \leq s \leq n$$

We then simulated the deployment of "virtual scanners" in the simulated area to generate encounters inferred from scanner observations. From the perspective of a scanner, two devices have an encounter when both devices are simultaneously within range of the scanner for a specified time period.

With these two data sets, we can then compare the baseline "actual" encounters with the inferred "scanner" encounters to understand the nature and frequency of encounter errors introduced by the use of scanners. The use of simulation also allows us to investigate the effects of scanner density on the accuracy and completeness of scanned data by deploying up to thousands of scanners per square kilometer.

### Mobility simulator

We employed a mobility simulator which implements the Lévy walk mobility model described in [RSH$^+$08]. We note that a considerable variety of synthetic mobility models have been proposed over time, including models proposed after the Lévy walk model (e.g., SLAW [LHK$^+$09], SWIM [KMS10], and individual-mobility [SKWB10]). We sidestep debates about the "best" mobility model, and instead observe that the Lévy model has the merits of validation with large realistic traces [RSH$^+$08] and is relatively popular and increasingly well understood (e.g., [LKC$^+$11]). Other models might result in different absolute values for malware propagation times and encounter distributions, but, given the inherent approach of using scanners to infer device encounters, we believe that the effects we observe are illustrative of the problem and not the mobility model.

We consider a number of agents, each carrying a device with a radio range of $10m$ (a typical range for proximity communication using Bluetooth), and we consider a pair of devices to be copresent if both are within the other's radio range. As before, we make the simplifying assumption that radios produce a sharply-demarcated disc of constant signal strength. The agents move in steps, with each step being comprised of a *flight* — motion in a single direction $\theta$ (randomly chosen from a uniform distribution such that $0^\circ \leq \theta \leq 360^\circ$) — followed by a *pause*, during which the agent is stationary. For each step, the flight length and pause time are chosen randomly from two Lévy distributions respectively having scale factors $\alpha$ (flight length) and $\beta$ (pause time). Additionally, for flight length and pause time values, we apply unit scale factors $c$ and $d$, and maximum values $t_f$ and $t_p$.

As in [RSH$^+$08] flight time (and hence velocity) is related to flight length to reflect the greater probability that longer flights use a mode of transportation other than walking. Flight length is given by $t_f = kl^{1-p}$, where $k$ and $p$ are constants and $0 \leq p \leq 1$. For flights of less than 500$m$, we use values of $k = 18.72$ and $p = 0.79$. For longer flights over 500$m$, we use values of $k = 1.37$, $p = 0.46$. We set unit scale factors for flight length $c$ of 10$m$, and pause time $d$ of 1 second in all simulations.

To reduce the effects of reflection on device mobility patterns, we define a large square area within which devices move (3000$m \times$ 3000$m$). If devices reach the edge of this area, their flights reflect off the outer boundary and continue their current flight step. We also define a smaller central inner area (1000$m \times$ 1000$m$) as the area of device interaction, and consider encounters between devices only within this area.

Inside the central inner area we deploy "virtual scanners" at fixed locations. These static scanners, like the mobile devices, have a 10$m$ radio range and are placed at least 20$m$ apart to avoid overlapping coverage areas. For simplicity of implementation, the scanners were placed on a square lattice, resulting in scanner coverage of 79% of the total area.

To investigate whether differing mobility parameters affected the extent and nature of errors in the scanned data, we use three sets of parameters for the scale factors of flight length ($\alpha$) and pause time ($\beta$) distributions in the Lévy walk model. Each of these three pairs of $\alpha$ and $\beta$ values represent simulation parameters found to fit well with empirical GPS datasets gathered from sets of walkers in three separate locations [RSH$^+$08]: San Francisco ($\alpha = 0.75$, $\beta = 1.68$), NCSU ($\alpha = 0.86$, $\beta = 0.99$) and KAIST ($\alpha = 0.97$, $\beta = 0.45$). For each of the three mobility parameter sets we performed 25 simulation runs lasting one week of simulated time, each for 900 devices. We assume all devices are susceptible, corresponding to malware propagation among mobile users who share devices with the same platform (and are a subset of all mobile users — as in Chapter 3). In each case, we deployed 2,500 scanners within the inner area (1000$m \times$ 1000$m$) of the simulation.

### 5.2.2 Simulation results

Our simulations produced datasets containing, for each mobility trace, a set of device encounters sensed by the mobile devices themselves, and a set of scanned encounters inferred by the “virtual scanners”. In comparing the two sets of encounters, our aim was to highlight the errors introduced by incorrect inferences leading to *partial match* and *spurious match* scanner encounters and their impact on simulations of malware propagation models using the encounter data.

**Comparing malware propagation dynamics**

As an initial test of our assertion that the use of scanned encounter data may lead to inaccurate estimates of malware propagation, we performed a simple malware propagation simulation using the encounter data from our simulator. At a high level, proximity-based malware propagation is a form of data dissemination in opportunistic ad-hoc networks. As such, since such malware propagation strongly depends upon the distributions of device contacts and contact durations, it is particularly useful for evaluating the sensitivity of such data dissemination applications to errors in device encounter data. As discussed above, our aim is to investigate the errors which arise in encounter data as a result of the scanners’ deviation from ideal behavior. A deployment of ideal scanners would infer a subset of device encounters, selected without bias, whose size is related to the proportion of area under scanner coverage.

Since scanner coverage in our simulation was incomplete, we would not expect propagation between the scanned and device encounters to match, even in the unlikely event that our virtual scanners behaved ideally. To control for the effects of incomplete scanner coverage, we created a normalized set of device encounters for use in our propagation model. This dataset consists of a subset sampled at random from the set of device encounters such that, for each mobility trace, the subset contains the same number of encounters as the corresponding set of scanned encounters. While we make no attempt to match the particular encounters taking place at scanner sites in this subset, we would expect the aggregate characteristics of the encounters to match those which our scanners would have inferred had they behaved ideally.

Figure 5.5 shows mean propagation over time over 100 runs on each mobility trace. Each simulation run assumed one initially infected device in a standard susceptible-infected (SI) model, with all devices susceptible and a latency for propagation of 30 seconds. For all three of the mobility parameter sets we see, as expected, that propagation proceeds more slowly in the “normalized” subset of device encounters.

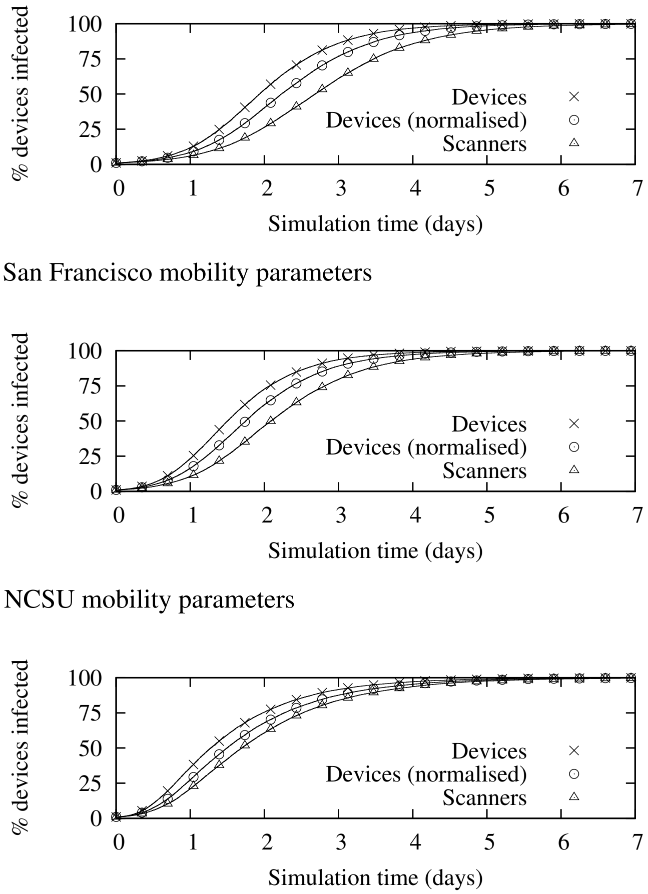
It is also apparent that when the scanned encounter sets are used in the model, despite having the same number of encounters as the normalized subsets of device encounters, propagation is slower still in all three mobility traces. The difference between propagation using the normalized device encounters and the scanner encounters is large. After two days of simulation time, the respective mean proportions of infected devices are: SF 40.2% vs. 26.4%, NCSU 61.6% vs. 46.4%, and KAIST 67.9% vs. 60.8%.

This experiment showed that the use of scanned encounters, when compared to subsets of device encounters, leads to an underestimation of propagation rates in proximity malware models. The deviation suggests that our scanners are not behaving ideally, and are introducing wrongly inferred *partial match* and *spurious match* encounters. The extent of these errors of inference is sufficient to alter the characteristics of the whole set of device encounters.

### Encounter overlaps

Our malware propagation simulation shows that the sets of scanned encounters differ from those detected directly by the mobile devices, and that this difference is attributable to incorrect inferences by the scanners which lead to the reporting of erroneous encounters. To better understand the nature of these erroneous encounters, we directly compared the scanner and encounter data from each individual mobility trace. By considering the encounters between each device pair which met (or was inferred to have met) at least once during the simulation, we show the proportion of encounter types present in the scanned encounter data (including *missed encounters*).

For each encounter between devices, we determined whether an encounter between the same two devices (in the set of scanned encounters) existed with the same start and end time (an *exact match*), or whether one or more *partial matches* existed which overlapped it in time. Device encounters where no match or overlap was found correspond to *missed encounters* (combining *missed-one* and *missed-two* static contacts).



**(a)** San Francisco mobility parameters

**(b)** NCSU mobility parameters

**(c)** KAIST mobility parameters

**Figure 5.5:** Malware propagation, comparing device encounters, normalized device encounters and scanned encounters.

Repeating the process from the perspective of the scanned encounters revealed the *spurious matches* which did not overlap any device encounters.

We first compared all encounters in the device and scanned encounter sets for each mobility parameter set (i.e., a minimum encounter duration of zero). We see that approximately two-thirds of device encounters have a corresponding *exact match* or *partial match* in the scanned encounters. There is little difference in this proportion between the three mobility sets. The incidence of exact matches is low, representing 6.5%, 5.5% and 4.4% of total device encounters in the San Francisco, NCSU and KAIST mobility parameters, meaning that almost all of the encounters actually inferred by the

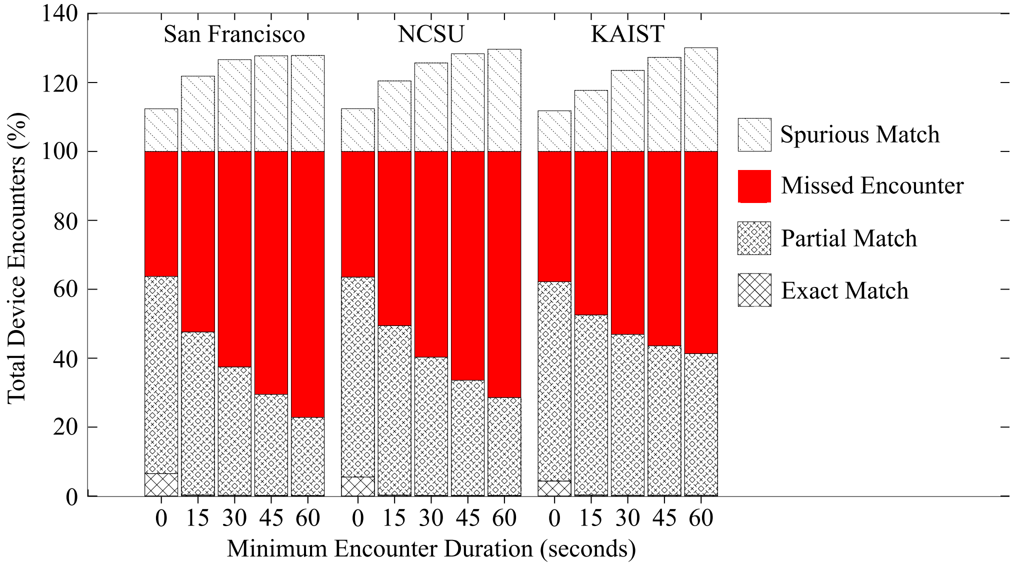


**Figure 5.6:** Relationships between device and scanner encounters by encounter type.

scanners are either *partial matches* or *spurious matches*.

Given the sensitivity of proximity-based malware propagation to encounter duration (since infection latency may be 30 seconds or more), we repeated the experiment using subsets of the device and scanner encounters with successively higher minimum durations. For all of the mobility traces, we first see that, once a minimum threshold of 15 seconds is imposed, the proportion of exact matches becomes vanishingly small. Where exact matches do occur, they are typically between encounters of short duration. This result is not unexpected, and suggests that very short encounters may simply offer less opportunity for erroneous inferences to take place.

We also see that the proportion of missed encounters and spurious encounters rises for all three mobility traces as the minimum encounter duration threshold increases. A calculation of correlation between the proportion of missed encounters to total device encounters and minimum latency threshold suggests that a strong relationship exists in all cases (SF $r^2 = 0.96$, NCSU $r^2 = 0.96$, KAIST $r^2 = 0.92$). A similar calculation of correlation between the proportion of spurious encounters and minimum encounter duration showed a less strong relationship which appeared to strengthen in the less diffusive mobility parameter sets (SF $r^2 = 0.71$, NCSU $r^2 = 0.82$, KAIST $r^2 = 0.89$).

The relationship between encounter length and proportion of missed encounters appears counter-intuitive. While practically all encounters over 15 seconds for all mobility traces do not match exactly between the scanner and device encounters, we had expected that longer encounters would experience a higher level of partial overlaps, if only by chance. Our experiments showing the opposite to be true suggest that there are differences between the distribution of encounter durations in the scanner and encounter datasets, with the scanner datasets simply including fewer long encounters to match against the device encounters.

### Encounter duration

The correlation between increasing encounter duration and incidence of missed and spuriously matched encounters in the scanner data led us to investigate the distribution of encounter duration between the device and scanner encounter sets. The duration of encounters is important when modeling the propagation of proximity-borne malware, where propagation between devices might occur only during uninterrupted connections of 30 seconds or more.

For each of the three mobility parameter sets, all simulation runs were combined to produce large sets of device encounters and scanner encounters. We calculated the proportion of encounters within each encounter set which were longer than a set of latency thresholds for proximity malware transmission. As Table 5.1 shows, in all cases a smaller proportion of the scanned encounters exceeds the latency thresholds. In other words, the scanned encounters underestimate the duration of the device encounters considerably. For longer but still realistic latency values of one minute, the proportion of scanned encounters lasting long enough to allow malware propagation to occur is lower than the device encounters by between 12% and 42% across the three sets of mobility parameters.

### Encounter “uniqueness”

We have shown that the scanned encounters generated by our simulator differ from the device encounters by having characteristics which lead to underestimation of malware propagation when these scanned encounters are used as source data. To further investigate the extent to which the scanned encounters differ from the device encounters,

**Table 5.1:** Encounters exceeding example malware propagation latencies (in seconds)

| | Device encs. (m) | Scanned encs. (m) | $\frac{P(D)}{P(S)}$ | $\frac{n(D)}{n(S)}$ |
|---|---|---|---|---|
| **SF** | | | | |
| All | 3.20 (100.0%) | 2.84 (100.0%) | 0.0% | -11.1% |
| >15 | 0.38 (11.9%) | 0.28 (10.0%) | -15.9% | -25.2% |
| >30 | 0.14 (4.3%) | 0.09 (3.2%) | -25.6% | -33.8% |
| >45 | 0.57 (1.8%) | 0.03 (1.2%) | -34.7% | -41.9% |
| >60 | 0.02 (0.7%) | 0.01 (0.4%) | -42.6% | -49.0% |
| **NCSU** | | | | |
| All | 2.53 (100.0%) | 2.24 (100.0%) | 0.0% | -11.4% |
| >15 | 0.42 (16.5%) | 0.32 (14.2%) | -13.8% | -23.6% |
| >30 | 0.17 (6.8%) | 0.12 (5.3%) | -22.1% | -31.0% |
| >45 | 0.80 (3.2%) | 0.05 (2.3%) | -28.4% | -36.6% |
| >60 | 0.04 (1.5%) | 0.02 (1.0%) | -33.2% | -40.9% |
| **KAIST** | | | | |
| All | 1.40 (100.0%) | 1.19 (100.0%) | 0.0% | -15.3% |
| >15 | 0.37 (26.1%) | 0.29 (24.0%) | -8.0% | -22.0% |
| >30 | 0.19 (13.9%) | 0.15 (12.4%) | -10.6% | -24.3% |
| >45 | 0.12 (8.6%) | 0.09 (7.6%) | -11.8% | -25.3% |
| >60 | 0.08 (5.7%) | 0.06 (5.0%) | -12.4% | -25.8% |

we compared the distribution of contact degrees of each set of encounter data. The distribution of contact degrees is a key driver of malware propagation: the dynamics of epidemic spread in networks with heavy-tailed distributions of encounter degrees differ significantly from “fully-mixed” models in which all agents are equally likely to meet [New02].

However, contact degree is related to encounter rates, and the incomplete coverage of the area provided by scanners will likely result in lower encounter rates. As a result the contact degrees of device and scanned encounters cannot be directly compared. To address this issue, we calculated a normalized metric, *encounter uniqueness*, which is the proportion of unique devices within the total devices encountered in a given period. In the case where a device meets each other device only once, all its encounters can be described as *unique*, giving a value of 1.0. As the proportion of encounters with previously-seen devices increases the ratio of unique encounters falls. For encounters with similar distributions of encounter duration, we would expect higher encounter uniqueness to correspond to increased rates of malware propagation.

To ensure comparability across our simulation data, we calculated the uniqueness values for encounters from the simulation start until each device in the simulation had met a given number of unique devices. We repeated this process for each of the mobility traces across all three mobility parameter sets. Figure 5.7 shows the distribution of the uniqueness ratio for the device and scanned encounters.

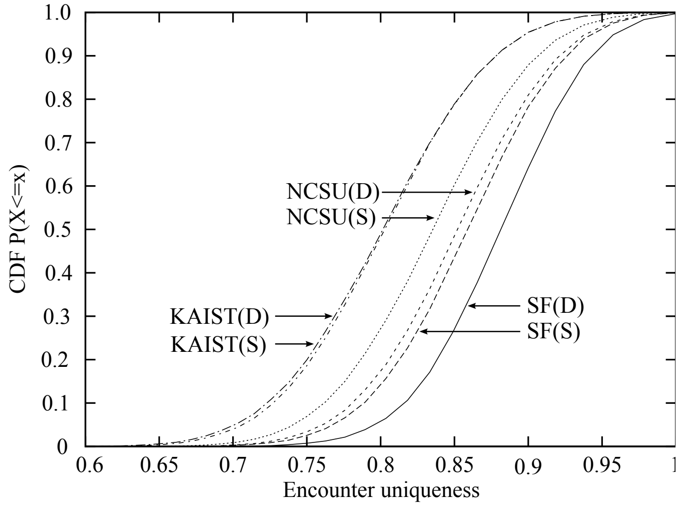


**Figure 5.7:** “Uniqueness” of device vs. scanner encounters

As expected, the more diffusive mobility parameter sets (NCSU, SF) show a higher encounter uniqueness. Longer flight lengths mean that devices are less likely to repeatedly encounter devices they have previously met. However, the two more diffusive mobility parameter sets are also most affected by underestimation of encounter uniqueness in the scanned encounters, while the least diffusive mobility parameter set (KAIST) shows very little difference in encounter uniqueness between the device encounters and the scanned encounters.

## 5.3 Summary

Our detailed examination of errors induced by inferring device encounters from third party scanners suggests caution in the use of such data sets, for instance, for the study of flooding-based data dissemination applications like proximity malware propagation. However, it is also suggestive of a potential way forward.

We have demonstrated the circumstances in which "bridging" errors between pairs of out-of-range devices occur. Further, we have shown, under assumptions of equal and homogeneous communication ranges, that over 41% of device encounters inferred from simultaneous scanner sightings were incorrect. In the case of mobile devices, these incorrect inferences have a complex effect on the accuracy of scanned encounters as they accumulate over time. As well as encounters which are missed or spuriously inferred by considering sightings at scanners, unreliable inference results in inferred encounters which have shorter durations than the actual encounters between devices, and underestimates the extent to which the devices encounter new, unmet devices.

The magnitude of these differences is sensitive to the underlying mobility characteristics of the devices being scanned, with more diffusive mobility correlating with increased errors. In all three sets of mobility parameters we tested (each closely matching GPS trails gathered from human movement), the extent of errors introduced through inferring copresence by simultaneous presence at scanners led to a significant underestimation of the rate at which proximity-based malware would spread amongst devices.

On the other hand, our use of a mobility simulator to compare actual encounters observed from mobility trails with encounters inferred from sightings at scanners suggests a method for mitigating erroneous inferences of copresence in data gathered by scanners deployed in the field. In cases of highly diffusive mobility, where the errors introduced by bridging appear to be most pronounced, the quality of scanned encounter data might be materially improved, leading to more accurate simulations of malware spread and countermeasures.

Estimated or observed characteristics of mobility patterns around the scanners, such as the distribution of velocities, flight lengths and pause times would be used to set initial parameters for a mobility simulator. This simulator would then be populated with virtual scanners similar to those used in the field, and used to infer simulated encounters.

The mobility parameters used in the simulation could then be improved iteratively until the simulated encounters closely matched the statistical properties of those gathered from the field scanners.

The malware propagation model could then be based on the direct encounters between devices in the simulator. Since the same fundamental geometry leads to errors in simulated scanners and the real deployed scanners we would expect the incidence of bridging errors in both cases to be similar, provided the simulator's mobility parameters closely match the observed characteristics of mobility around the deployed scanners. This being the case, the direct encounters between devices in the simulator should capture the observed properties of human mobility at the scanner sites, while reducing errors from incorrect inferences — and in doing so be closer to the real human encounters which took place around the scanners.

Chapter 5 is based on the material as it partly appears in the "Proceedings of the 4th International Conference on Communication Systems and Networks 2012" with the title "Limitations of scanned human copresence encounters for modelling proximity-borne malware" by James Mitchell, Eamonn O'Neill, Gjergji Zyba, Geoffrey M. Voelker, Michael Liljenstam, András Méhes, Per Johansson.

# Chapter 6

# Conclusion

The alarming growth in mobile malware developed during recent years motivates extensive studies to better understand its unique properties and to effectively protect users. In this dissertation I examined the unique characteristics of mobile malware propagation along with the effectiveness of various defense strategies. Below I summarize the contributions of our research and indicate the impact as well as future directions.

## 6.1 Contributions

I studied the propagation of mobile malware that spreads through direct pairwise communication mechanisms, such as Bluetooth or WiFi, between devices in geographic proximity. Compared to previous studies, I incorporated both theoretical analysis and simulation using synthetic and real world traces to validate our results. I showed that proximity malware can infect all the susceptible devices of a campus-size area in a matter of a few days. Furthermore, I applied realistic parameter sets regarding human mobility to generate encounters between devices. Malware propagation with these encounters showed that the more diffusive mobility is, the faster malware propagates within an area.

While proximity malware spreads through direct encounters between devices, it remains completely unobserved by the operator, making defense against it challenging. Thus, defense must start with local detection and then can continue with collaboration between the devices themselves or the mobile operator. I evaluated, for the first time, the propagation of proximity malware looking at three defense strategies: local detection alone, local detection with collaboration among the devices themselves, and local detection in collaboration with the mobile operator. When such defense mechanisms are applied, I found that local dissemination of signatures can be effective in containing the propagation, while global coordinated strategies (where the mobile operator is involved) contain it even faster.

I also studied the impact of user social behavior on proximity malware propagation. From real world traces I identified areas where people congregate, split the population into two groups based on social behavior, and labeled them as frequent or transient visitors. Even though transient visitors are considered unimportant, as they appear very little compared to frequent users, they compose the majority of the population in many real world scenarios. Our results of malware propagation simulation using either set or the entire population showed that transient visitors can be surprisingly effective at spreading the malware. Our analysis can be applied also for any data dissemination application and, to our knowledge, this is the first study that considers transient visitors for data propagation.

Proximity malware propagation is significantly impacted by the direct pair-wise encounters between devices. Such encounters are difficult to collect, however without being intrusive to the user. A popular method is to infer them through scanners. We extensively studied the strengths and limitations of deploying static scanners for inferring direct pair-wise encounters, and found that scanner-inferred encounters exhibit significant statistical differences compared to the actual encounters. Particularly, the set of encounters inferred from scanners differs from the actual encounters in terms of duration distribution and probability of encountering previously unmet devices. I believe these discrepancies should be considered by future research. In terms of proximity malware propagation, I showed that scanned encounters yield a slower propagation compared to actual encounters for devices with the same mobility characteristics.

## 6.2 Impact and Future Directions

In our studies I focused on proximity malware propagation in areas of limited size (e.g., university campus) and found that malware can infect the population of such areas in a few days. However, the scope of these analyses should be expanded to look at larger areas. Of particular concern is proximity malware that strategically uses the network operator to combine long-distance infections with local propagation. Rather than flooding the network with traffic, thereby raising alarms, such hybrid malware could use infrequent communication to minimize chances of detection.

Our study showed that proximity-based signature dissemination effectively limits worm propagation. Due to the stronger signature assumptions, though, the broadcast-based signature dissemination strategy gives the best results. An antivirus server, by having more information and more processing power, is able to generate better quality signatures and instruct the devices on how to remove the malicious code. The centralized nature of this strategy has also two significant advantages. First, it can better handle polymorphic worms by having copy of the malicious code from different devices. Second, it is a very attractive solution for antivirus/operator companies, as they can provide this feature as an additional chargeable service, or generally provide better security to their customers. We should note, though, that all the above techniques are susceptible to false positives.

I looked at how the social behavior of users impacts malware propagation when malware spreads only by using a subset of the devices (Vagabonds versus Socials) in a single area. A number of interesting research directions follow, including studying the characteristics of inter-area malware propagation, the dynamics of user social behavior (e.g., Vagabonds becoming Socials in other areas), and the interactions between Vagabonds and Socials in supporting malware dissemination.

Our detailed analysis of errors which are generated by inferring device encounters through static scanners suggests caution in the use of such data sets directly. However, the scanner-captured data could be used indirectly to further improve the accuracy of simulations of malware spread and countermeasures. First, estimated or observed characteristics of mobility patterns around the scanners, such as the distribution of velocities, flight lengths, and pause times could be used to set initial parameters for a mobility simulator, which in turn generates encounters between simulated devices and inferred encounters using virtual scanners. Then, iterative improvement would be possible by matching inferred encounter properties from the real world data set and encounter properties from virtual scanners. After this process, direct encounters from simulated devices could be used for the desired analysis.

In my work I have shown the potential threats of proximity malware propagation, as well as the impact of various defense strategies on containing its propagation. Yet, mobile malware is still in its infancy. Given the pervasive adoption of mobile devices throughout the world, the threat of mobile malware will continue to increase and my work provides guidance to future efforts tackling such threats.

# Bibliography

[3GP] 3GPP. Multimedia Broadcast Multicast Service. http://www.3gpp.org/ftp/Specs/archive/23_series/23.246. 27

[BB08] Michel Benaïm and Jean-Yves Le Boudec. A Class Of Mean Field Interaction Models for Computer and Communication Systems. *Performance Evaluation*, 65(11–12):823–838, May 2008. 58

[BHSP08] Abhijit Bose, Xin Hu, Kang G. Shin, and Taejoon Park. Behavioral Detection of Malware on Mobile Handsets. In *Proceedings of the 6th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 225–238, Breckenridge, CO, June 2008. 10, 12

[BRB+08] Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti. Exploiting Self-Reported Social Networks for Routing in Ubiquitous Computing Environments. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob)*, pages 484–489, Avignon, France, October 2008. 11

[Can12] Canalys. Smart phones overtake client pcs in 2011. http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011, February 2012. 1

[CHC+05] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Lab, Cambridge, UK, February 2005. 12

[CHC+06] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–13, Barcelona, Spain, April 2006. 36, 38, 45, 47

[CMMP08] Paolo Costa, Cecilia Mascolo, Mirco Musolesi, and Gian-Pietro Picco. Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 25(5):748–760, June 2008. 11

[CMZ07] Luca Carettoni, Claudio Merloni, and Stefano Zanero. Studying Bluetooth Malware Propagation: The BlueBag Project. *IEEE Security and Privacy*, 5(2):17–25, March–April 2007. 13

[DG99] Daryl J. Daley and Joseph Mark Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, Studies in Mathematical Biology, Cambridge, UK, 1999. 20

[DH07] Elizabeth M. Daly and Mads Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 32–40, Montréal, QC, September 2007. 11

[EP06] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, March 2006. 12, 13, 36

[ETMP05] William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the 12th ACM conference on Computer and Communications Security (CCS)*, pages 393–404, Alexandria, VA, November 2005. 2

[FFC$^+$11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steven Hanna, and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, pages 3–14, Chicago, IL, October 2011. 1

[FLJ$^+$07] Chris Fleizach, Michael Liljenstam, Per Johansson, Geoffrey M. Voelker, and András Méhes. Can You Infect Me Now? Malware Propagation in Mobile Phone Networks. In *Proceedings of the 5th ACM Workshop on Recurring Malcode (WORM)*, pages 61–68, Alexandria, VA, November 2007. 14

[FS] F-Secure. F-Secure Virus Information Pages: Commwarrior. http://www.f-secure.com/v-descs/commwarrior.shtml. 2, 14, 16

[GHB08] Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. 10, 12, 18, 43

[GLZC09] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Multicasting in Delay Tolerant Networks: A Social Network Perspective. In *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 299–308, New Orleans, LA, May 2009. 11

[HCY08] Pan Hui, Jon Crowcroft, and Eiko Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 241–250, Hong Kong, China, May 2008. 11, 36, 37, 39

[HKA08] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. *Computer Networks*, 52(14):2690–2712, October 2008. 38

[HSL10] Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre. Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, pages 866–874, San Diego, CA, March 2010. 11, 37, 39

[ICM09] Stratis Ioannidis, Augustin Chaintreau, and Laurent Massoulié. Optimal and Scalable Distribution of Content Updates over a Mobile Social Network. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1422–1430, Rio de Janeiro, Brazil, April 2009. 11

[KKK06] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006. 12

[KMS10] Sokol Kosta, Alessandro Mei, and Julinda Stefa. Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking. In *Proceedings of the 7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, Boston, MA, June 2010. 75

[KOP$^+$10] Vassilis Kostakos, Eamonn O'Neill, Alan Penn, George Roussos, and Dikaios Papadongonas. Brief encounters: Sensing, modeling and visualizing urban mobility and copresence networks. *ACM Transactions on Computer-Human Interaction*, 17(1):1–38, March 2010. 13

[Kos07] Vassilis Kostakos. Experiences with urban deployment of Bluetooth (given at UCSD). http://www.ee.oulu.fi/~vassilis/files/presentations/pres_ucsd.pdf, March 2007. 9, 17

[KSS08] Hahnsang Kim, Joshua Smith, and Kang G. Shin. Detecting Energy-Greedy Anomalies and Mobile Malware Variants. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, pages 239–252, Breckenridge, CO, June 2008. 10

[Laba] Kaspersky Lab. IT Threat Evolution: Q1 2013. http://www.securelist.com/en/analysis/204792292/IT_Threat_Evolution_Q1_2013. 8

[Labb] Kaspersky Lab. Mobile Malware Evolution: An Overview, Part 2. https://www.securelist.com/en/analysis?pubid=201225789. 7

[Labc] Kaspersky Lab. Mobile Malware Evolution: An Overview, Part 3. https://www.securelist.com/en/analysis?pubid=204792080. 7

[Labd] Kaspersky Lab. Mobile Malware Evolution: An Overview, Part 4. https://www.securelist.com/en/analysis/204792168/Mobile_Malware_Evolution_An_Overview_Part_4. 7, 8

[Labe] Kaspersky Lab. Trojan.SymbOS.Mosquit.a. https://www.securelist.com/en/descriptions/103141/Trojan.SymbOS.Mosquit.a. 7

[Labf] Kaspersky Lab. Virus.Win9x.CIH. https://www.securelist.com/en/descriptions/old19775. 6

[Labg] Kaspersky Lab. Worm.SymbOS.Cabir.a. https://www.securelist.com/en/descriptions/old60663. 2, 7, 14, 16

[Ley09] Michael Ley. Does the Knee in a Queuing Curve Exist or is it just a Myth? http://www.cmg.org/measureit/issues/mit61/m_61_16.html, July 2009. 41

[LHK+09] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. SLAW: A Mobility Model for Human Walks. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pages 855–863, Rio de Janeiro, Brazil, April 2009. 75

[LKC+11] Kyunghan Lee, Yoora Kim, Song Chong, Injong Rhee, and Yung Yi. Delay-Capacity Tradeoffs for Mobile Networks with Lévy Walks and Lévy Flights. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, pages 3128–3136, Shanghai, China, April 2011. 75

[McA] McAfee. Securing Mobile Devices: Present and Future. http://www.mcafee.com/us/resources/reports/rp-securing-mobile-devices.pdf. 8

[MCL+08] Abderrahmen Mtibaa, Augustin Chaintreau, Jason LeBrun, Earl Oliver, Anna-Kaisa Pietiläine, and Christophe Diot. Are You Moved by Your Social Network Application? In *Proceedings of the 1st ACM SIGCOMM*

*Workshop on Online Social Networks (WOSN)*, pages 67–72, Seattle, WA, August 2008. 11

[MGC+07] Andrew G. Miklas, Kiran K. Gollu, Kelvin K.W. Chan, Stefan Saroiu, Krishna P. Gummadi, and Eyal de Lara. Exploiting Social Interactions in Mobile Systems. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, pages 409–428, Innsbruck, Austria, September 2007. 11

[MM06] Mirco Musolesi and Cecilia Mascolo. A Community Based Mobility Model for Ad Hoc Network Research. In *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN)*, pages 31–38, Florence, Italy, May 2006. 13

[MMDA10] Abderrahmen Mtibaa, Martin May, Chistophe Diot, and Mostafa Ammar. PeopleRank: Social Opportunistic Forwarding. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, pages 111–115, San Diego, CA, March 2010. 11, 37, 39, 45

[MN05] James W. Mickens and Brian D. Noble. Modeling Epidemic Spreading in Mobile Environments. In *Proceedings of the 4th ACM Workshop on Wireless Security (WiSe)*, pages 77–86, Cologne, Germany, September 2005. 9, 12, 25

[Mob] NQ Mobile. 2011 Mobile Security Report. http://docs.nq.com/2011_NQ_Mobile_Security_Report.pdf. 8

[MPS+03] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. The Spread of the Sapphire/Slammer Worm. Technical report, CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE, San Diego, CA, January 2003. 6

[MSN05] Mehul Motani, Vikram Srinivasan, and Pavan S. Nuggehalli. PeopleNet: Engineering a Wireless Virtual Social Network. In *Proceedings of the 11th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 243–257, Cologne, Germany, August 2005. 11

[MV05] Marvin McNett and Geoffrey M. Voelker. Access and Mobility of Wireless PDA Users. *Mobile Computing and Communications Review (MC2R)*, 9(2):40–55, April 2005. 36, 38

[New02] Mark E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, July 2002. 82

[Nie12] Nielsen. Smartphones account for half of all mobile phones, dominate new phone purchases in the us. http://blog.nielsen.com/nielsenwire/online_ mobile/smartphones-account-for-half-of-all-mobile-phones-dominate-new-phone-purchases-in-the-us, March 2012. 1

[NW07] Kenrad E. Nelson and Carolyn Williams. *Infectious Disease Epidemiology: Theory and Practice*. Jones and Bartlett Publishers, 2007. 57

[OKK$^+$06] Eamonn O'Neill, Vassilis Kostakos, Tim Kindberg, Ava Fatah gen. Schiek, Alan Penn, Danaë Stanton Fraser, and Tim Jones. Instrumenting the City: Developing Methods for Observing and Understanding the Digital Cityscape. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp)*, pages 315–332, Orange County, CA, September 2006. 13

[PD09] Anna-Kaisa Pietiläinen and Christophe Diot. Experimenting with Opportunistic Networking. In *Proceedings of the 4th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Kraków, Poland, June 2009. 40

[POL$^+$09] Anna-Kaisa Pietiläinen, Earl Oliver, Jason LeBrun, George Varghese, and Christophe Diot. MobiClique: Middleware for Mobile Social Networking. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, pages 49–54, Barcelona, Spain, August 2009. 11

[PSDG09] Michal Piórkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. A Parsimonious Model of Mobile Partitioned Networks with Clustering. In *Proceedings of the 1st International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pages 1–10, Bangalore, India, January 2009. 39

[Reg] The Register. Zombie PCs spew out 80% of spam. http://www.theregister.co.uk/2004/06/04/trojan_spam_study/. 6

[Res12] ABI Research. $389 million mobile application security market set to explode as threats increase significantly. http://www.abiresearch.com/press/389-million-mobile-application-security-market-set/, September 2012. 1

[RN08] Christopher J. Rhodes and Maziar Nekovee. The opportunistic transmission of wireless worms between mobile devices. *Physica A: Statistical Mechanics and Its Applications*, 387(27):6837–6844, June 2008. 9

[RSH$^+$08] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, and Song Chong. On the Levy-walk Nature of Human Mobility. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, pages 924–932, Pheonix, AZ, April 2008. 10, 12, 15, 18, 19, 22, 24, 75, 76

[RSLC07] Injong Rhee, Minsu Shin, Seongik Hongand Kyunghan Lee, and Song Chong. Human Mobility Patterns and Their Impact on Delay Tolerant Networks. In *Proceedings of the 6th ACM Workshop on Hot Topics in Networks (HotNets)*, Atlanta, GA, November 2007. 36

[SCM$^+$06] Jing Su, Kelvin K. W. Chan, Andrew G. Miklas, Kenneth Po, Ali Akhavan, Stefan Saroiu, Eyal de Lara, and Ashvin Goel. A Preliminary Investigation of Worm Infections in a Bluetooth Environment. In *Proceedings of the 4th ACM Workshop on Recurring Malcode (WORM)*, pages 9–16, Alexandria, VA, November 2006. 9, 12, 13, 16, 17, 63

[Sec] Second Life. http://www.secondlife.com. 40

[SFM09] SFMTA. San francisco transportation fact sheet. http://www.sfmta.com/cms/rfact/documents/SFFactSheet2009_November2009_FINAL.pdf, November 2009. 39

[SKWB10] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, September 2010. 75

[SQBB10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, February 2010. 12

[Tro] Trojan.SymbOS.Skuller.a. Trojan.SymbOS.Skuller.a. https://www.securelist.com/en/descriptions/108177/Trojan.SymbOS.Skuller.a. 7

[VPDB08] Matteo Varvello, Fabio Picconi, Christophe Diot, and Ernst Biersack. Is There Life in Second Life? In *Proceedings of the 4th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pages 1–12, Madrid, Spain, December 2008. 40

[VV10] Matteo Varvello and Geoffrey M. Voelker. Second Life: a Social Network of Humans and Bots. In *Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pages 9–14, Amsterdam, Netherlands, June 2010. 40

[VYC05] Michail Vlachos, Philip Yu, and Vittorio Castelli. On Periodicity Detection and Structural Periodic Similarity. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM)*, pages 449–460, Newport Beach, CA, April 2005. 42

[WE06] Nicholas Weaver and Dan Ellis. White Worms Don't Work. *;login:*, 31(6):33–38, December 2006. 27

[YE06] Guanhua Yan and Stephan Eidenbenz. Bluetooth Worms: Models, Dynamics, and Defense Implications. In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC)*, pages 245–256, Miami Beach, FL, December 2006. 9, 10

[YE07] Guanhua Yan and Stephan Eidenbenz. Modeling propagation dynamics of bluetooth worms. In *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 42–51, Toronto, Canada, June 2007. 9, 12

[YFC$^+$07] Guanhua Yan, Hector D. Flores, Leticia Cuellar, Nicolas Hengartner, Stephan Eidenbenz, and Vincent Vu. Bluetooth Worm Propagation: Mobility Pattern Matters! In *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 32–44, Singapore, March 2007. 9, 25

[YHC07] Eiko Yoneki, Pan Hui, and Jon Crowcroft. Wireless Epidemic Spread in Dynamic Human Networks. In Pietro Liò, Eiko Yoneki, Jon Crowcroft, and Dinesh C. Verma, editors, *Bio-Inspired Computing and Communication, 1st Workshop on Bio-Inspired Design of Networks (BIOWIRE), Revised Selected Papers*, pages 116–132, Cambridge, UK, April 2007. 12

[YHCC07] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks. In *Proceedings of the 10th ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 225–234, Chania, Greece, October 2007. 11

[ZB07] Hui Zang and Jean C. Bolot. Mining Call and Mobility Data to Improve Paging Efficiency in Cellular Networks. In *Proceedings of the 13th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 123–134, Montréal, QC, September 2007. 12

[ZLG06] Hui Zheng, Dong Li, and Zhuo Gao. An Epidemic Model of Mobile Phone Virus. In *Proceedings of the 1st International Symposium on Pervasive Computing and Applications (SPCA)*, pages 1–5, Xinjiang, China, January 2006. 9