PRACTICAL PRIVACY AND SECURITY FOR OPPORTUNISTIC NETWORKS

Iain Parris

**A Thesis Submitted for the Degree of PhD
at the
University of St Andrews**

2014

Practical privacy and security for opportunistic networks

Iain Parris

This thesis is submitted in partial fulfilment for the degree of PhD
at the
University of St Andrews

August 2014

Abstract

When in physical proximity, data can be directly exchanged between the mobile devices people carry — for example over Bluetooth. If people cooperate to store, carry and forward messages on one another's behalf, then an opportunistic network may be formed, independent of any fixed infrastructure.

To enable performant routing within opportunistic networks, use of social network information has been proposed for social network routing protocols. But the decentralised and cooperative nature of the networks can however expose users of such protocols to privacy and security threats, which may in turn discourage participation in the network.

In this thesis, we examine how to mitigate privacy and security threats in opportunistic networks while maintaining network performance. We first demonstrate that privacy-aware routing protocols are required in order to maintain network performance while respecting users' privacy preferences. We then demonstrate novel social network routing protocols that mitigate specific threats to privacy and security while maintaining network performance.

Acknowledgements

I would like to give special thanks to my supervisor, Tristan Henderson, who has consistently gone above and beyond in providing guidance and advice — in addition to help, patience and even culinary critiques.

My thanks also go to people with whom I have worked in the School of Computer Science, and in St Andrews in general. In particular, I would like to mention Saleem Bhatti and the members of the former STEAL group: Fehmi, Devan, Greg, Yi, Saray and Markus. I would also like to thank Ishbel Duncan for her advice as second supervisor throughout (and especially while writing-up); the CS administration team, in particular Alex and Paula; the fixit team; Ruth Unsworth; and all of my reviewers: Mike Weir, Mike Livesey, Colin Allison, and Tom Kelsey.

Beyond St Andrews, it has been a pleasure to collaborate within the PVNets project. I would also like to give thanks for the flexibility and encouragement from my work colleagues — including Ron Muir, Simon Campbell, Scott Thomson, and especially Jose Vazquez. And I am lucky to have the support of my friends, with particular mention for David Baker, Svenja Gosen, Ray Danbakli, Tiffani Sanders, Brittani Walker, Mahasin Ameen, and two Andrews (both Beymer and Cardall).

I would like to express my deepest thanks to my family too. This includes my parents, John and Ruhy; my sisters, Claire and Dawn; my grandparents, Shahla, Big Dad, Granny Audrey and my late Grandpa; and my wider family on all sides — Parris; Mottahed, with special mention for Mama Shirazi; French, with special mention for Russell, Cathy, Donnie and Buddy; and most recently Kingham.

Finally, I would like to thank Brandi, for absolutely everything. And just for her, I will even note Tegan.

Collaboration statement

While generally written using the plural voice (“we”), this thesis has been written by me, and the research presented in the standard chapters of this thesis has been principally performed by me.

Earlier versions of the research presented in Chapters 4–6 has been presented in peer-reviewed multi-author publications, where I am the first author. As is customary, my PhD supervisor (Tristan Henderson) is a co-author of each publication. He provided supervision for the experimental research, and participated in the paper writing process (although any text and figures also used in this thesis is my own). The following is a list of these publications, including a description of the nature of the collaborations involved where there are additional co-authors:

- Chapter 4
    - Iain Parris and Tristan Henderson. ***The impact of location privacy on opportunistic networks.*** *Proceedings of the Fifth IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC)*, Lucca, Italy, June 2011. IEEE Computer Society Press.
    - Iain Parris, Fehmi Ben Abdesslem, and Tristan Henderson. ***Facebook or Fakebook? The effect of simulation on location privacy user studies.*** *Proceedings of the Privacy and Usability Methods Pow-Wow (PUMP)*, Dundee, UK, September 2010. British Computer Society.
        * Fehmi Ben Abdesslem was the lead researcher for collecting the LocShare dataset, but I also contributed significantly (approximately equally) towards implementation and execution. The analysis presented, however, is my own.

  - Iain Parris, Fehmi Ben Abdesslem, and Tristan Henderson. ***Facebook or Fakebook? The effects of simulated mobile applications on simulated mobile networks.*** *Ad Hoc Networks*, 12:35-49, January 2014.
    - ∗ As for the previous publication, I collaborated with Fehmi Ben Abdesslem to collect the LocShare dataset. The analysis presented is, again, my own.

- Chapter 5
  - Iain Parris, Greg Bigwood, and Tristan Henderson. ***Privacy-enhanced social network routing in opportunistic networks.*** *Proceedings of the Second IEEE International Workshop on SEcurity and SOCial Networking (SESOC)*, pages 624–629, Mannheim, Germany, March 2010. IEEE Computer Society Press.
    - ∗ Greg Bigwood prepared the augmented SASSY dataset trace, as detailed in Appendix A.1.1. The remainder of the work presented is my own.
  - Iain Parris and Tristan Henderson. ***Privacy-enhanced social-network routing.*** *Computer Communications*, 35(1):62-74, January 2012.
- Chapter 6
  - Iain Parris and Tristan Henderson. ***Friend or Flood? Social prevention of flooding attacks in mobile opportunistic networks.*** *Proceedings of the Sixth International Workshop on Hot Topics in Peer-to-peer computing and Online Social neTworking (HotPOST)*, Madrid, Spain, June 2014.

A full list of all my publications from during my PhD studies — including those more loosely related to the thesis and/or where I am not the first author — is provided in Chapter 1.3.

Declarations

Candidate's declarations

I, Iain Parris, hereby certify that this thesis, which is approximately 30,000 words in length, has been written by me, and that it is the record of work carried out by me or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in September 2008 and as a candidate for the degree of Doctor of Philosophy in September 2008; the higher study for which this is a record was carried out in the University of St Andrews between 2008 and 2014.

date ____________________ *signature of candidate* ____________________

Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

date ____________________ *signature of supervisor* ____________________

Permission for publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any *bona fide* library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

PRINTED COPY

No embargo on print copy

ELECTRONIC COPY

No embargo on electronic copy

date ______________________ *signature of candidate* ______________________

date ______________________ *signature of supervisor* ______________________

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
– Tim Peters, The Zen of Python

But it's time to exorcise these demons;
[They're] doing jumping jacks now.
. . .
Let me remind you of what got me this far:
Picture me quitting, now draw a circle around it
and put a line through it.
– Marshall Mathers, Not Afraid & Survival

Contents

Chapter 1

Introduction

People are increasingly carrying wireless communication devices, such as mobile phones, during their daily lives. Currently, communication between these devices relies upon the existence of fixed infrastructure, such as mobile phone networks or Wi-Fi access points. But this means that in infrastructure-less scenarios, where infrastructure is unavailable or otherwise undesirable to use (e.g., due to cost), then communication is not possible.

These devices may alternatively communicate with one another in an ad hoc manner: two devices may directly exchange messages when in physical proximity, via wireless protocols such as Bluetooth or Wi-Fi Direct, without requiring any fixed infrastructure. If many such devices, acting as network nodes, cooperate to carry each other's messages, then a decentralised *opportunistic network* is formed, in a disconnected store-carry-and-forward architecture.

The decentralised and cooperative nature of such networks, where there may be no traditional infrastructure and where nodes are expected to cooperate and forward data for each other, can however expose the network users to privacy and security threats. This in turn may discourage participation in the network.

This thesis examines the following research questions:

Q1: How can we determine the performance impact of changes in behaviour due to users' privacy concerns, when we do not have a deployed opportunistic network?

Q2: Can privacy and security concerns that arise through the use of opportunistic networks be mitigated through cooperative social behaviour, while maintaining network performance?

1.1 Thesis

We offer the following thesis statement:

> Privacy and security threats within opportunistic networks can be mitigated through cooperative social behaviour, without reducing network performance.

To support this thesis, we make three research contributions. We demonstrate that:

1. If opportunistic network users reduce their participation in the network due to privacy concerns, then this may impact the network performance.
2. Privacy can be preserved for network participants, while maintaining performance, through use of modified routing protocols.
3. Social network information may be used to mitigate a security threat, where a malicious attacker floods the network with messages.

1.2 Outline

The remainder of the thesis is structured as follows.

Chapters 2 and 3 outline related research, and the current state of the art.

- Chapter 2 describes the research background for opportunistic networks. This includes outlining how opportunistic networks relate to other kinds of network; how social networks may be used to enable performant opportunistic network routing; and privacy and security threats to which opportunistic network users may be exposed.

- Chapter 3 summarises research related to opportunistic network privacy and security, including the current state of the art.

Chapters 4, 5 and 6 document the analysis and work carried out in support of the stated thesis.

- Chapter 4 demonstrates that privacy concerns may impact network performance, should opportunistic network users reduce their participation in the network due to a perceived privacy threat.
- Chapter 5 details modified routing protocols, demonstrating that privacy may be preserved while maintaining routing performance.
- Chapter 6 demonstrates that a threat to security — a flooding attack — can be mitigated when using a social network routing protocol, while maintaining routing performance.

Finally, Chapter 7 concludes with a summary of the research contributions of this thesis, and discusses potential research that may be performed in the future.

1.3 Publications

I have contributed to the following publications during the course of my research. In those publications for which I am listed as the first author, the main contributions in design, analysis and implementation are my own; otherwise I have contributed but the main contributions are those of the first author as shown. In each case, I gratefully acknowledge the contributions of my fellow authors.

1. Iain Parris. ***Privacy-enhanced opportunistic networks.*** *Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp),*

pages 213–214, Pisa, Italy, February 2010. ACM Press. Extended abstract. doi:10.1145/1755743.1755794

2. Iain Parris, Greg Bigwood, and Tristan Henderson. ***Privacy-enhanced social network routing in opportunistic networks.*** *Proceedings of the Second IEEE International Workshop on SEcurity and SOCial Networking (SESOC)*, pages 624–629, Mannheim, Germany, March 2010. IEEE Computer Society Press. doi:10.1109/PERCOMW.2010.5470511

3. Iain Parris, Fehmi Ben Abdesslem, and Tristan Henderson. ***Facebook or Fakebook? The effect of simulation on location privacy user studies.*** *Proceedings of the Privacy and Usability Methods Pow-Wow (PUMP)*, Dundee, UK, September 2010. British Computer Society. Online at: http://scone.cs.st-andrews.ac.uk/pump2010/papers/parris.pdf

4. Fehmi Ben Abdesslem, Iain Parris, and Tristan Henderson. ***Mobile experience sampling: Reaching the parts of Facebook other methods cannot reach.*** *Proceedings of the Privacy and Usability Methods Pow-Wow (PUMP)*, Dundee, UK, September 2010. British Computer Society. Online at: http://scone.cs.st-andrews.ac.uk/pump2010/papers/benabdesslem.pdf

5. Iain Parris and Tristan Henderson. ***The impact of location privacy on opportunistic networks.*** *Proceedings of the Fifth IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC)*, Lucca, Italy, June 2011. IEEE Computer Society Press. doi:10.1109/WoWMoM.2011.5986149

6. Iain Parris and Tristan Henderson. ***Practical privacy-aware opportunistic networking.*** *Proceedings of the Doctoral Consortium at the 25th BCS Conference on Human-Computer Interaction (British HCI Doctoral Consortium)*, pages 553-557. Newcastle upon Tyne, UK, July 2011. British Computer Society. Online at: http://ewic.bcs.org/content/ConWebDoc/45174

7. Fehmi Ben Abdesslem, Iain Parris, and Tristan Henderson. ***Reliable online social network data collection.*** In Ajith Abraham and Aboul-Ella Hassanien, editors, *Computational Social Networks: Mining and Visualization*, volume 3 of *Springer Computer Communications and Networks Series*, chapter 8, pages 183-210. Springer-Verlag, London, UK, 2012. doi:10.1007/978-1-4471-4054-2_8

8. Iain Parris and Tristan Henderson. ***Privacy-enhanced social-network routing.*** *Computer Communications*, 35(1):62-74, January 2012. doi:10.1016/j.comcom.2010.11.003

9. Iain Parris, Fehmi Ben Abdesslem, and Tristan Henderson. ***Facebook or Fakebook? The effects of simulated mobile applications on simulated mobile networks.*** *Ad Hoc Networks*, 12:35-49, January 2014. doi:10.1016/j.adhoc.2012.05.008

10. Iain Parris and Tristan Henderson. ***Friend or Flood? Social prevention of flooding attacks in mobile opportunistic networks.*** *Proceedings of the Sixth International Workshop on Hot Topics in Peer-to-peer computing and Online Social neTworking (HotPOST)*, Madrid, Spain, June 2014. Online preprint: http://ip.host.cs.st-andrews.ac.uk/publications/hotpost2014.pdf

Chapter 2

Background

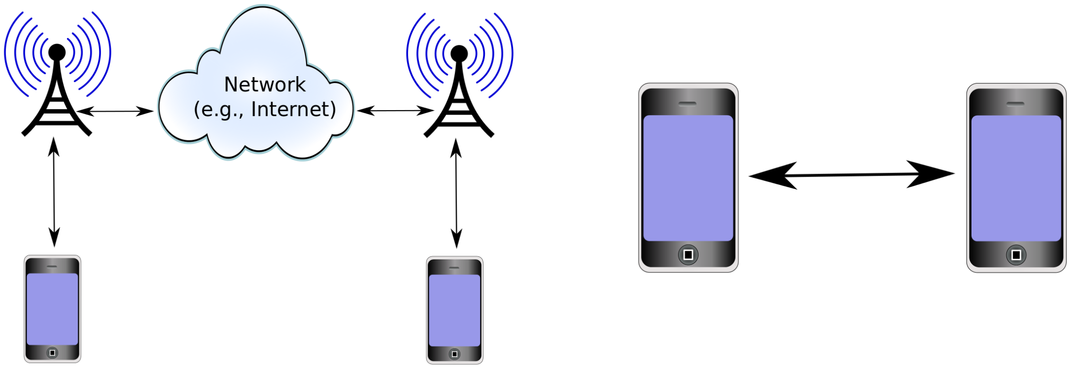
In this chapter, we discuss the research background. We begin by motivating opportunistic networking research, outlining its origins in relation to other wireless networking research. We then describe the routing challenge within opportunistic networks, and give an overview of how social network information may be used to enable performant routing. Finally, we discuss some of the intrinsic security and privacy threats to which users of opportunistic networks may be exposed.

2.1 The evolution of opportunistic networks

Increasing numbers of people carry mobile devices — such as mobile phones, laptops, and tablet computers — during their daily lives. These devices may participate in wireless networks.

Commonly-deployed wireless networks currently require fixed infrastructure. For example, devices may utilise cellular networks, which requires fixed cellular radio towers (Figure 2.1(a)). Or devices may connect via IEEE 802.11 (Wi-Fi) infrastructure mode, which requires fixed access points.

In scenarios where fixed infrastructure does not exist, or does exist but is unattractive to use (e.g., due to being unreliable, congested, or prohibitively expensive), then it is possible for mobile devices to communicate directly with one another



(a) Traditional network with infrastructure. Phones communicate only via fixed infrastructure.

(b) Without infrastructure. Phones can communicate directly with one another while in physical proximity.

Figure 2.1: Mobile phones communicating with and without fixed infrastructure.

peer-to-peer, via physical layer protocols such as Bluetooth or Wi-Fi Direct (Figure 2.1(b)).

We summarise the areas of wireless networking research focused on creating logical wireless networks on top of such physical protocols, beginning with mobile ad hoc networks (MANETs).

2.1.1 Mobile ad hoc networks

Mobile ad hoc networks (MANETs) are self-configuring wireless networks of mobile devices. Messages can only be passed directly between two devices — or *nodes* — when they are *neighbours*, i.e., when they are in sufficiently-close physical proximity for the physical layer wireless protocol. The network is formed through the cooperation of nodes, with each node available to act as a router to forward messages on behalf of other nodes — as in a mesh network. This enables non-neighbouring nodes, which are too distant for direct communication with one another, to be able to exchange messages, with messages passed hop-by-hop through the network. MANETs may be viewed as a generalisation of wireless mesh networks, where all nodes are mobile. As a consequence of node mobility, MANET network topology is dynamic: as nodes physically move, their neighbours may change as they leave proximity of previous neighbouring nodes and enter proximity of new neighbouring nodes. MANET routing protocols have therefore been developed to take account of the dynamic network topology.

There are a wide variety of MANET routing protocols [72]. These protocols fall into two major categories:

Proactive (table-driven) In proactive routing protocols, each node maintains a regularly-updated routing table of routes for every other node in the network, via background exchange of routing control messages.

Routing tables are maintained at every node, for all other nodes, irrespective of whether data messages are exchanged between a given pair of nodes. This allows low-latency usage of any given route, at the expense of background routing control message overhead even when data messages are not being sent.

An example protocol of this type is the Optimized Link State Routing Protocol (OLSR) [39]. Nodes discover their immediate neighbours via regular *Hello* messages, and receive lists of their neighbours' neighbours (i.e., nodes two hops away) via their neighbours' responses. Each node's local view of the network topology is flooded throughout the network — via an optimisation of using a set of elected nodes, *multipoint relays*, to reduce the number of redundant *topology control* messages sent throughout the network. Each node can thus construct a view of the global network topology, and can locally determine optimal routing paths to every other node.

Reactive (on demand) In reactive routing protocols, routes are found on demand at the time of use. When a node wishes to send data to another node, it may invoke a *route discovery* process, to find a route to that node. This increases the latency for first-use of a new route, but eliminates routing control message overhead during times when the network is silent.

Example protocols of this type are Ad-hoc On-Demand Distance Vector (AODV) routing [115] and Dynamic Source Routing (DSR) [80].

In AODV, to establish a new route to a destination node, a *route request* message is first broadcast by the sender node. This message is retransmitted and flooded through the network by intermediate nodes, each appending their identifier to every retransmitted copy of the message to note the path taken through the network. Once the message is received by the intended destination, a *route reply* message is sent along the reverse path, and local routing tables at each node along the reverse path cache the route. This results in multiple temporary routes being established, and for subsequent data exchange each forwarding node may choose which among these temporary routes to use, usually on the basis of the fewest hops to the destination.

In DSR, route request and route reply messages are sent similarly to AODV. But unlike AODV, each message is source routed — i.e., the route that a message should take through the network is specified by the sender alongside the message. This means that a discovered route need only be "remembered" by the sender, and intermediate nodes need not maintain any local routing state.

A combination of proactive and reactive routing may be used within a single routing protocol. For example, with the Zone Routing Protocol (ZRP) [67] each node maintains a predefined *zone* of nodes within a fixed number of hops, and uses proactive routing within the zone. Reactive routing is used to communicate with nodes outside the zone.

A fundamental assumption underlying all MANET routing protocols is that *an end-to-end route must exist between source and destination*. If no such route exists, then the network is partitioned and communication between the two endpoints is not possible.

A different approach to networking is needed when this assumption does not hold; this led to the creation of *delay tolerant networks*.

2.1.2 Delay Tolerant Networks

Delay (or Disruption) Tolerant Networks (DTNs) enable networking even in scenarios where we relax the following assumptions [57]:

- End-to-end paths exist between all nodes.

- The round-trip time between any pair of nodes is small.
- Packet loss is low.

DTNs were originally proposed for interplanetary communications [32]. A communication link between planets necessarily involves high delay, due to the large distances (of the order of light-minutes). Additionally, the link may only be operative at certain times, due to orbital mechanics and requiring line-of-sight.

More generally, DTNs include networks where some links may be high-delay, i.e., where the network may frequently (or perpetually) experience long-duration partitioning [77]. For example, in a military scenario, two units on a battlefield may be able to communicate internally within each unit via traditional MANET protocols, but be out of range for direct communication between units. Communication by each unit with a helicopter may intermittently be possible as the helicopter passes overhead [114].

In order to enable networking in these conditions, DTNs employ a store-and-forward strategy. Messages are stored at DTN nodes until the intermittent link is operational, and then forwarded.

For example, RFC 4838 [34] defines a message-oriented overlay, the *bundle layer*. The network is split into regions, with regions separated by intermittent links. Traditional protocols (e.g., MANET routing protocols) are used internally within regions. Between regions, across the intermittent links, store-and-forward bundle protocols are used.

Such protocols enable networking in the example DTNs above, where there are relatively few intermittent links. But what about in the case where *all* links may be intermittent? This is the premise for *opportunistic networks*.

As a note on terminology: the terms *DTN* and *opportunistic network* are often used interchangeably in the research literature. Following [114], we regard opportunistic networks as a subset of DTNs, where all links are subject to high delay. Figure 2.2 illustrates the relationship between opportunistic networks and DTNs, in the context of the other types of network that we have discussed.

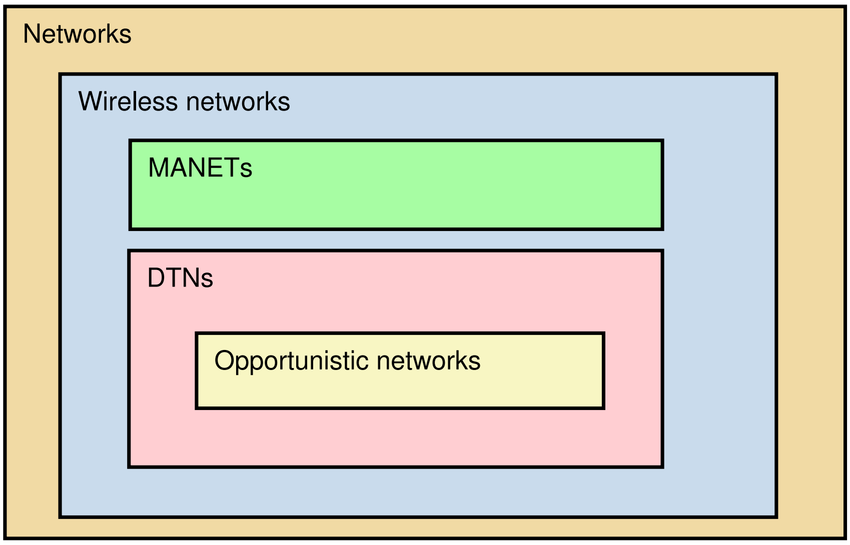


Figure 2.2: Venn diagram illustrating opportunistic networks in relation to other network types.

2.1.3 Opportunistic networks

Opportunistic networks [114] are essentially disconnected MANETs, which utilise a store-carry-and-forward paradigm. Mobile nodes opportunistically exchange messages with one another during encounters, i.e., when they are in physical proximity. The exchanged messages are then stored and carried by each node, and may be opportunistically forwarded to other nodes during future encounters. As in MANETs, each node is mobile; as in DTNs, no contemporaneous end-to-end path need ever exist between source and destination.

This leads to the defining feature of opportunistic networks: *the network topology is anticipated to change whilst each message is in flight*. As a consequence of this, there is no global knowledge of network topology. Each node must therefore rely on only local knowledge in order to make routing decisions, i.e., to choose whether or not to forward each carried message during each opportunistic encounter with another node.

Example special cases of opportunistic networks include:

Wildlife monitoring Small mobile sensing devices may be attached to animals. Sensed data may be opportunistically shared between devices when animals are in proximity to one another, to create the network and thus enable low-cost non-intrusive wildlife monitoring. For example, in ZebraNet custom collars were attached to zebras [82].

Message ferrying Special *message ferries* — mobile nodes with non-random movement patterns — may be used to opportunistically "ferry" data within a network [154]. For example, buses may store-and-forward data between remote communities.

Vehicular networks (VANETs) Vehicles may contain embedded devices, able to opportunistically communicate with one another as part of the network whenever two vehicles pass within physical proximity [31].

Throughout this thesis, however, our focus when discussing opportunistic networks is those networks formed by humans carrying personal mobile devices, such as mobile phones, during their daily lives. In a term coined by Hui *et al.*, such opportunistic networks are also known as *Pocket Switched Networks* [74]. There are not yet any real-world large-scale deployments of such networks, although small-scale research prototypes have been implemented, such as for the Haggle Project [135].

Such networks can be used to create new applications — such as social media or information dissemination [45] — even in the absence of existing infrastructure, and even in disconnected scenarios where traditional MANETs would fail.

In the next section, we discuss routing approaches for these opportunistic networks.

2.2 Opportunistic network routing

We have seen that opportunistic networks present challenges for routing: network topology is dynamic, and so each node must rely on only local knowledge in order to make routing decisions during their encounters with other nodes.

The efficiency and performance of an opportunistic network depend on accurately determining which encountered nodes will be useful in forwarding. In this section, we survey routing protocols introduced to enable performant opportunistic network routing. We first discuss methods used to evaluate routing protocol performance, and then introduce example routing strategies.

2.2.1 Evaluating routing performance

As discussed in Section 2.1.3, there are not yet any any real-world large-scale opportunistic network deployments. The primary method used to evaluate the performance of opportunistic network routing protocols is therefore *trace-driven simulation*.

The traces used in such simulations are time-ordered lists of encounters between nodes, i.e., personal mobile devices. These traces may be *synthetic* (generated artificially by an analytic model of human mobility), or may be based on empirically-measured real-world human movements. Given such an *encounters list*, routing protocols may be simulated and thus directly compared, based on a variety of performance metrics.

Metrics commonly used for evaluating the performance of opportunistic network routing protocols include [75]:

Delivery ratio The proportion of messages that are successfully delivered, out of the total number of unique messages created.

Delivery delay The length of time taken for a message to reach its destination destination: the wall-clock time elapsed between when the message is first sent and when it first arrives at its intended final destination.

Opportunistic network routing protocols may involve a trade-off between performance, as measured by each of these metrics, and *delivery cost*, where the delivery cost is a measure of how many messages (including redundant messages) are used in order to deliver each message created. To formalise the notion of delivery cost, it may be defined as the total number of messages (including duplicates) transmitted, normalised by the total number of unique messages created.

2.2.2 Routing protocols

Examples of opportunistic network routing protocols include:

Epidemic [141] Message copies are forwarded between nodes during each and every encounter. Since messages are forwarded out along all paths, this approach indeed ensures (in ideal conditions, e.g., the absence of energy constraints) that, if a path exists between source and destination, the message will certainly find and follow this path to be delivered as quickly as possible. But sending large numbers of redundant messages, as epidemic routing is apt to do, is wasteful, and will drain the batteries of the mobile devices rapidly. This approach is optimal for delivery ratio and delivery delay performance, but very poor for delivery cost.

Direct delivery [141] Some variations of epidemic routing aim to reduce delivery cost — potentially at the expense of delivery ratio and delivery delay — by setting a maximum hop count for forwarding paths. In the limit, where the maximum hop count is one, the message may only be delivered directly from source to destination. This is the trivial case of opportunistic networking, where no intermediate forwarding nodes are used. This is optimal for delivery cost, since necessarily no redundant messages will be generated, but very poor for delivery ratio and delivery delay.

Spray and Wait [131] The goal of Spray and Wait is to bound the maximum number of copies of each message throughout the entire opportunistic network, i.e., to set a fixed upper limit for delivery cost. The source "sprays" a fixed maximum number of message copies into the network, by forwarding the message to the first nodes it encounters. Then each of these intermediate nodes "waits", and will only deliver the message directly to its destination. This protocol can be thought of as similar to epidemic routing with a maximum hop count of two, but also imposes a limit on the number of intermediate nodes to which the source may forward message copies.

MaxProp [31] MaxProp can be thought of as an optimisation of epidemic routing, to better cope with non-ideal conditions where transfer duration or network buffers may be limited. In these non-ideal conditions, nodes may not be able to exchange copies of all carried messages during encounters. MaxProp defines an order for message transmission, prioritising messages by sending them earlier during encounters, and therefore maximising the number of prioritised messages exchanged during a limited encounter. Message prioritisation is based on heuristics — for example messages with low hop count are prioritised.

PRoPHET [95] The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) aims to reduce delivery cost, i.e., unnecessary message replication. This is achieved by relying on the assumption that previous encounters between nodes indicate an increased probability of future encounters between the same nodes, i.e., encounters are non-random. Each node maintains an estimated probability — *delivery predictability* — for every destination, which is increased during each encounter with the destination node but decreased ("aged") with time. Message copies are forwarded between nodes only when the newly-encountered node has a higher delivery predictability for the destination than the node currently carrying the message, i.e., when the newly-encountered node is believed to be more likely to deliver the message to its destination.

RAPID [9] The Resource Allocation Protocol for Intentional DTN routing (RAPID) can, like MaxProp, be considered to be an optimisation of epidemic routing to prioritise messages, in order to better cope with the non-ideal bandwidth-limited condition where not all messages may be exchanged during encounters. The goal of RAPID is to optimise routing performance as measured by one explicitly-chosen performance metric, e.g., delivery ratio or delivery delay. This is achieved by prioritising messages to forward based on an explicit utility function, intended to estimate performance as measured by the chosen performance metric.

The routing protocols described above make use of network or mobility characteristics. Another strategy is to use *social network information* — information about social relations between the people who carry the mobile devices which make up the network — to inform routing decisions. This is the premise for *social network routing*.

2.2.3 Social network routing

In the context of social network routing, social networks are essentially graphs of social ties between nodes (users). In contrast to the opportunistic network topology — which, as discussed in Section 2.1.3, is highly dynamic, even on the timescales of typical message delivery — the social network graph is relatively stable, and infrequently updated.

The underlying assumption behind social network routing is that *encounters are more likely between devices carried by users who have a social relation* [43, 129]. The intuition behind the assumption is that people who know one another are more likely to make arrangements to meet (e.g., arranging a social activity), and/or that people who meet often may be more likely than random to again meet in future (e.g., work colleagues or commuters sharing a bus). The social network information may then be used in an attempt to inform more efficient routing decisions (i.e., to enable routing with high delivery ratio, low delivery cost, and low delivery delay).

There are two major categories of social network that may be utilised by social network routing protocols: *detected social networks (DSNs)* and *self-reported social networks (SRSNs)* [20].

DSNs are social networks *detected* by devices, based on logging historical encounters with other devices. The assumption is that if a device has frequently been encountered before, then it may likely be encountered again. DSNs are also sometimes referred to as “contact networks” [129].

SRSNs are social networks that have been manually recorded by users, to indicate their social relations. For example, on the popular *Facebook*[1] website, people explicitly note their social relations, as their *friends*. The assumption in social

[1] http://www.facebook.com

network routing is that friends (social relations, i.e., one-hop neighbours on the social network graph) are more likely to physically encounter one another than random nodes.

An advantage of utilising DSNs is that *any* frequently-encountered device may be used for forwarding, even if belonging to a person not necessarily known to the message sender. For example, two people sharing a part of a regular daily commute may be "familiar strangers" — not personally knowing one another, yet encountering one another frequently [102]. DSNs could detect this scenario, and exploit the regularity of encounters for opportunistic network forwarding. On the other hand, DSNs necessarily require a warm-up period in order to detect regularity of encounters. SRSNs, in contrast, while not able to exploit such "familiar strangers", are available to be used immediately upon joining the network. An additional possibility is to combine use of both DSNs and SRSNs, in order to gain the advantages of both: SRSNs can be used for bootstrapping the network, while after a warm-up period DSNs can be used to base decisions on historical encounter data.

A variety of social network routing protocols have been proposed, using social networks as categorised above. Examples include:

Bubble Rap [75] Bubble Rap utilises DSNs. Encounter patterns are analysed to detect *communities*, which are groups of nodes likely to meet frequently. Each node is assigned to at least one community. Additionally, the *centrality* of each node (both globally, and for each community to which it belongs) is measured; this is a metric for how likely the node is to be on the shortest path between two other nodes. The node centrality is also found to correlate with how "popular" the node is, such that a node which frequently encounters many other nodes has a high centrality.

Messages are first *bubbled* up, by being forwarded to encountered nodes if the encountered node is more "popular" (i.e., has higher global centrality) than the current node. Once the message reaches a node in the same community as the destination, the message is then forwarded only within the community (now exchanged during each encounter if the encountered node is in the same community as the destination, and has a higher local community centrality than the current node) until it reaches its destination.

The intuition is that the most likely nodes to use for forwarding are initially those which are more popular (and well-connected) globally, until a node within the same community as the destination is found. Then the communities represent groups of nodes which are likely to encounter one another frequently, and so the message should only be forwarded within the community until it reaches its destination.

SimBetTS [47] SimBetTS also uses DSNs, in a similar approach to Bubble Rap. Further metrics from social network analysis are employed, namely *betweenness centrality* and *social similarity*.

The betweenness centrality of a node is similar to the centrality used in Bubble Rap; it is a measure for how well-connected each node is within the network. The social similarity of two nodes is their number of *common neighbours*, i.e., the number of one-hop neighbouring nodes that they have in common.

These social network analysis metrics are used to dynamically calculate a utility value for each message during each encounter. The message is forwarded if the utility of the encountered node is higher than the current node. The intuition is that messages are likely to first be routed towards more central nodes, and then towards more similar nodes — this is similar to Bubble Rap, although social similarity is used in place of detected communities.

Habit [99] Habit combines usage of a DSN of frequently-encountered nodes (termed a "regularity graph") with a SRSN recording who is interested in receiving content from whom (termed an "interest graph"), to create an information-dissemination system. Nodes exchange local views of both social networks during encounters, such that each node maintains views of both graphs.

Messages are then source routed, with senders finding paths through the regularity graph that minimise overhead in the interest graph — i.e., minimise the number of nodes on the interest graph that are used for routing information which they are not also interested in receiving in their own right. During encounters, messages are forwarded according to the previously-chosen route.

Friendlist-based Social Forwarding [128] Friendlist-based Social Forwarding (FSF) utilises both DSNs derived from encounters and SRSNs from Facebook friendships for routing.

During each encounter, nodes exchange social network information. Similarly to SimBetTS, a utility value is dynamically calculated for each message — but in FSF this utility is based on combining two forms of social network information with equal weights: DSN node centrality and SRSN tie strength. The message is forwarded if the utility of the encountered node is higher than that of the current node.

PeopleRank [104] PeopleRank uses an SRSN of reported social ties between nodes, to determine the "importance" of each node within the social network. By analogy to Google's PageRank algorithm [27], nodes are ranked as "important" when they are socially linked to many other "important" nodes.

During encounters, messages are only forwarded if the encountered node has a higher PeopleRank rank than the current node. The intuition is that more socially well-connected nodes are better to use for forwarding.

SRSN Routing [20] SRSN Routing (SRSNR) uses SRSNs for a form of source routing. Senders include, with each message, a list of nodes which may be suitable for use in routing — the "friends" of the source node.

During each encounter, a given message is forwarded to the encountered node only if that node is on the list of candidate forwarding nodes specified by the source node, alongside the message. This forwarding occurs even if the two nodes are not themselves friends with one another.

The intuition is that encounters between the source node's friends are more likely than encounters between random nodes. For example, co-workers within the same building may both be friends of the source node, even if not necessarily each other, but may regularly encounter one another in hallways at work.

Table 2.1 summarises these six social network routing schemes, categorising each by the type of social network used. In this thesis our focus when discussing social network routing is primarily SRSN Routing (SRSNR), where SRSNs are used for source routing.

SNR protocol	Uses DSNs	Uses SRSNs
Bubble Rap [75]	Yes	No
SimBetTS [47]	Yes	No
Habit [99]	Yes	Yes
Friendlist-based Social Forwarding [128]	Yes	Yes
PeopleRank [104]	No	Yes
SRSN Routing [20]	No	Yes

Table 2.1: Social network type(s) used within social network routing schemes.

2.3 Security and privacy problems

Opportunistic networks are fundamentally decentralised, relying on participation by nodes in the network to forward and route messages for one another. But if all nodes are expected to act as routers, and some nodes may be untrusted, a variety of possible threats are introduced that may not exist in a traditional infrastructure network. Such threats may relate to the security of the network, or to the privacy of the users of the network.

We outline examples of some of the high-level security and privacy threats in this section. Mitigating these threats is a key area of opportunistic networking research [42], which we discuss further in Chapter 3.

2.3.1 Privacy

Potentially-private information which might be disclosed, or potentially be inferred, through participation in an opportunistic network includes:

Payload confidentiality Message contents may be disclosed, enabling messages to be read by unintended parties. This threat may be mitigated with encryption, if the sender and destination are able to agree on encryption keys [127].

Communication patterns Messages may be traced as they progress through the opportunistic network, allowing an attacker to infer communication patterns [92].

The attacker need not necessarily be able to read the plaintext message content in order to perform this attack: eavesdropping on multiple encounters, and being able to detect that the same message was transmitted during each encounter (e.g., with transmission of the same encrypted text, or through matching headers), would be sufficient.

Anonymity Whether a particular person is participating in an opportunistic network — and if so the link between their network identity and their real-life identity — may itself be considered private information. Anonymity is also linked to communication patterns: determination of the real-life identity of a pair of communicating nodes is a related privacy threat.

Location privacy Locations of the participants may be inferred from the messages which their mobile devices carry — whether in absolute terms ("Alice is at the supermarket"), or relative terms of co-location ("Alice and Bob were in the same location this afternoon"). This threatens location privacy, i.e., "the ability to prevent other parties from learning one's current or past location" [16]. Since messages are exchanged only when participants are in physical proximity, threats to the privacy of communication patterns (e.g., eavesdropping on messages progressing through the network) can enable inference attacks against location privacy (see Chapter 4.2).

Social graph privacy Social network information is utilised by social network routing protocols (see Chapter 2.2.3). This information may be leaked via the routing scheme, if the routing scheme shares such information with other nodes during routing. Social ties between users, and more generally the social graph, may be considered sensitive information, however [70, 89]. In routing schemes such as SRSN Routing, since social links are directly used to inform routing decisions, social graph privacy is additionally linked to communication patterns and location privacy: eavesdropping messages progressing through the network can allow inferences to be made about both (co-)location and social ties. We discuss this further in Chapter 4.2.

2.3.2 Security

Malicious nodes within opportunistic networks may perform attacks, which may significantly reduce the performance or availability of the network [37].

We note that many of these same high-level attacks are also possible in MANETs, since the basic idea behind each attack is a misbehaving node [30].

A non-exhaustive sample of attacks to which opportunistic networks are susceptible includes:

Black hole [49] A malicious node may discard any message which it has accepted, without forwarding the message in accordance with the routing protocol.

Opportunistic networks have a natural resilience to this attack, if using a replica-based routing protocol: since multiple copies of the message would typically exist in the network, a malicious node discarding one copy may have a more limited impact than in other network types.

Sybil [51] A Sybil Attack consists of forging multiple identities. Without a centralised authority to certify identity, a single malicious node may be able to spoof arbitrarily many identities within the network. For example, if an opportunistic network consisted of nodes identified solely by MAC address, then a single malicious node could appear to be any arbitrarily-high number of other nodes over time, by spoofing many MAC addresses.

This attack may undermine redundancy — for example, if a limited number of copies of the message are to be spread into the network, then a malicious node could accept all copies under different identities and then perform a black hole-style attack. Reputation-based approach may also be affected, since a Sybil node may assume other identities to falsely "vouch for" itself, or may bypass a blacklist-based system by changing to use a new identity once flagged.

Routing information falsification [30] If nodes in a network are expected to exchange routing information with one another, then a malicious node may falsify the information it supplies to any encountered node. For example, some of the social network routing protocols discussed in Section 2.2.3 involve exchanging social network information. This may impact routing performance, by distorting the other node's view of the network. This may also be combined with the Sybil attack, to spoof information about multiple false identities.

Flooding [144] A malicious node may flood the network with messages. If the identity of the node can be spoofed (as in the Sybil attack described previously), then this flooding may be untraceable. As the available resources of participating devices (e.g., battery) are finite, and may be drained by receiving and retransmitting these messages, this flooding attack may therefore act as a denial-of-service attack against participating network nodes.

2.4 Summary

In this chapter, we have discussed the evolution of opportunistic networks, from previous wireless network research. We have noted the following points:

- Opportunistic networks are an evolution of previous wireless networks, to enable networking in disconnected networks, i.e., even where no end-to-end path necessarily ever exists between source and destination.
- To avoid flooding the network, routing protocols have been developed that, in the absence of global knowledge at each node, utilise social network information.
- Simulation is the primary method used to evaluate routing protocol performance.
- Opportunistic networks using such routing protocols have inherent privacy and security threats for participants.

In the next chapter, we discuss the current research being performed to address these problems.

Chapter 3

Privacy and security in opportunistic networks

In this chapter, we survey current research on opportunistic networks, with a focus on privacy and security. We begin by discussing the accuracy and credibility of opportunistic network simulation under assumptions about potential users' real-world behaviour, including privacy-preserving behaviours. We then examine existing proposed methods to preserve users' privacy in opportunistic networks. Finally, we consider defences against security attacks to the availability of opportunistic networks.

3.1 Simulation

We noted in Chapter 2 that there do not currently exist real-world large-scale opportunistic network deployments. It is also often impractical to create a deployment for experimental purposes — both in general on a large-scale, and even on a smaller scale when constant refinement and development of protocols are required. Performance evaluation is therefore primarily achieved through simulation. How accurate, or credible, are these simulations?

Network type	Nodes are mobile	Is delay-tolerant
Mesh network	No	No
Wireless sensor network	No	Yes
VANET	Yes (vehicles)	Yes
MANET	Yes (people)	No
Opportunistic network	Yes (people)	Yes

Table 3.1: Wireless network properties compared and contrasted (as referenced throughout this thesis).

3.1.1 Simulating the network

Simulation has been widely used as a tool for network research in general, beyond only opportunistic networks, to aid development of new types of network and protocols. Many researchers have therefore investigated the problem of how to create credible network simulations across a wide variety of network types — including traditional infrastructure networks such as the Internet [58] and telecommunications networks [112], as well as wireless networks more closely related to opportunistic networks such as mesh networks [136], wireless sensor networks [62], VANETs [24] and MANETs [4]. (As a note on terminology, there is not universal agreement in the research literature about delineations between these wireless network types; for clarity, Table 3.1 compares and contrasts the properties of such networks as referenced within this thesis.)

Challenges for credible network simulation have been widely noted. For example, in a seminal paper Pawlikowski *et al.* survey over 2,200 publications containing telecommunications network simulation studies, and note a "crisis of credibility" [112]. More recently, Kurkowski *et al.* [91] and Hiranandani *et al.* [71] survey MANET MobiHoc publications, for 2000–2005 and 2006–2010 respectively. Both surveys find that many simulations described are not credible, or not repeatable. Joerer *et al.* [79] report a similar finding for VANET simulations.

To improve the situation, and produce more credible network simulations, various guidelines and best practices have been proposed. For example, Pawlikowski *et al.* focus on two necessary conditions for a credible simulation study: "appropriate pseudo-random generators ... and appropriate analysis of simulation output data" [112]. Andel and Yasinsac [4] survey MANET simulation credibility, and also provide general recommendations — including to document all simulation settings to allow repeatability, to perform an appropriate number of independent runs, and where possible to validate simulations against a real-world implementation. Perrone and Ward [116] have introduced a framework ("SAFE") to aid automated application of such general simulation best practices.

In addition to general simulation guidelines, improving wireless network simulation credibility in particular is also well-researched. Measurement-based comparisons between simulations and testbeds have been made for networks as diverse as wireless mesh networks [136], wireless sensor networks [62] and MANETs [142]. Physical-layer wireless assumptions have also been considered. For example, Newport *et al.* [106] and Anderson *et al.* [5] experimentally test physical-layer assumptions often used in wireless simulations. These assumptions include fixed-distance circular transmission range (two nodes can communicate with one another if and only if they are within a set distance of one another), and symmetry (if node A can successfully transmit to node B, then node B can successfully transmit a reply back to node A). They demonstrate that these assumptions may not hold, which may impact results of simulations making these assumptions.

As a subtype of wireless networks, these well-studied network-layer issues are also applicable to opportunistic network simulation. For example, Bittencourt *et al.* discuss a virtualisation-based software testbed for more accurately simulating wireless channel characteristics in opportunistic network simulation [22]. But an additional consideration for credible simulation of opportunistic networks is simulating the network users.

3.1.2 Simulating users

In infrastructureless networks — including opportunistic networks, MANETs and VANETs — the network is formed by mobile nodes carried by network users. This means that the network performance is dependent on the behaviour of the users, and thus user behaviour must be considered as a factor in creating credible network simulations.

The most well-studied facet of user behaviour in the context of credible network simulations is mobility. Messages can be exchanged only when nodes are in physical proximity to one another, and so network performance strongly depends on user movements [35, 86, 33]. A variety of approaches for simulating mobility have been proposed, which can be grouped into two broad categories: purely synthetic, or based on empirical observations.

A commonly-used example of a purely synthetic mobility model is the *Random Waypoint Mobility Model* [81, 71], where nodes travel between random locations, pausing upon arriving at each new destination. A danger, however, with such purely synthetic models is that movements may be unrealistic, leading to potentially unreliable network performance simulation results [148, 76]. The alternative is to base node movements on empirical observations. One approach, used for example by Schwamborn *et al.* [124], is to construct a more realistic mobility model based on empirical data. Another approach, used by Kim *et al.* [88], is to use real mobility traces to drive simulations on an emulated testbed. A third approach, used widely in opportunistic network simulations, is to perform contact-driven simulation: traces of encounters between devices (also known as "contact traces") may be used to simulate message-passing routing protocols realistically, since knowing encounters (even if lacking full mobility data) are sufficient for such simulations [130, 134, 10].

In addition to mobility, other factors too may also affect the credibility of opportunistic network simulations. Ristanovic *et al.* compare opportunistic network performance of a real testbed to trace-driven simulation, and note that assumptions such as infinite buffers for storing messages may affect simulation performance results [120]. Mota *et al.* also note that assumptions made for traffic patterns may not match real opportunistic network applications [103]. Where social network information is used for routing, a further factor to consider is the realism of the social network itself; Orman and Labatut, for example, study the impact of social network realism on community detection algorithms [108].

Another type of user behaviour that may affect opportunistic network performance is privacy-preserving behaviour. Participating in an opportunistic network may expose users to privacy threats, as detailed in Chapter 2.3.1. Due to these threats, users may become less willing to participate in message forwarding on behalf of other users [101]: during times when the opportunistic network application is not running on their devices, a given user's exposure to privacy threats is minimised, since the user is effectively invisible to the network.

To create a credible opportunistic network simulation, we may therefore need to consider how to measure and model such privacy-preserving user behaviour, and its corresponding impact on the performance of the network. We highlight this as a gap not addressed in current opportunistic network simulation research.

3.2 Privacy attack defences

As we have discussed previously in Chapters 2.3.1 and 3.1, users of a naïve opportunistic network may be exposed to privacy threats, which can cause them concern [101]. In this section, we outline current research related to adapting opportunistic networks to preserve users' privacy.

We consider methods proposed to protect each of the categories of potentially-private information that were identified in Chapter 2.3.1: payload confidentiality, communication patterns, anonymity, location privacy, and social graph privacy.

3.2.1 Payload confidentiality

As in other networks, one approach to protect the confidentiality of message payloads is by employing encryption. Traditional approaches to key distribution and key management, however, are difficult to apply to opportunistic networks. Due to the disconnected nature of the network, with delays inherent to opportunistic network message delivery, querying a public key infrastructure (PKI) is likely to be impractical [55, 85], and even much of the research on key distribution in MANETs does not generalise to disconnected opportunistic networks [25].

A possible exception — noted as potentially being suitable for disconnected networks such as opportunistic networks [125, 8, 127, 85] — is in the use of identity-based cryptography (IBC) [26]. In IBC, node identifiers serve as their public keys. Private keys are obtained by each node from a globally-trusted third party, the "private key generator" (PKG). This key generation step need only be performed once; after initially obtaining this private key from the PKG, a node need not be able to communicate again with the PKG while participating in the opportunistic network. IBC has two main drawbacks, however, in that a globally-trusted third party is required, and forward secrecy is lacking (i.e., a stolen private key allows decryption of all prior communication with that key). Seth and Keshav [125] partially address these problems, by describing a time-based and hierarchical IBC scheme, with a hierarchy of PKGs descending from a root PKG. But even this solution requires a globally-trusted third party, which may be a limiting requirement for general opportunistic networks.

An alternative approach to ensure payload confidentiality is by routing opportunistic network messages via trusted social contacts. For example, Bulut and Szymanski [29] describe a two-period routing protocol: although messages are unencrypted, routing is preferentially via trusted friends where possible, to reduce the risk of an untrusted node receiving the unencrypted payload. El Defrawy *et al.* [55] describe a routing protocol where messages are sent encrypted via social contacts, under the assumption that social contacts know one another's encryption keys.

Summarising, if sender and receiver are able to agree on encryption keys — which may be challenging in an opportunistic network — then encryption may be used to preserve message confidentiality. We note this privacy threat primarily for completeness; our focus is on eavesdropping, metadata privacy, and inference attacks.

3.2.2 Communication patterns and anonymity

Traffic analysis may reveal communication patterns — for example, determination of those nodes participating in the opportunistic network, or with which other people users communicate (sender-receiver linkability). Onion routing [64] is a popular proposed solution [92, 126, 78, 1]. The idea behind onion routing is to mix and route messages via intermediate nodes, with successive layers of encryption "peeled away" at each hop, in order to provide anonymity. Tor is the most widely-known implementation of onion routing on the traditional Internet.

In contrast to traditional onion routing implementations, where messages are routed along a fixed path of intermediate nodes, adaptations of onion routing for opportunistic networks — such as PEON by Le *et al.* [92] and ARDEN by Shi *et al.* [126] — often generalise onion routing to allow messages to be forwarded by intermediate groups of nodes, rather than fixed intermediate nodes. This provides more flexibility for routing, since requiring a message to be sent along a single fixed path would be a strong requirement for an opportunistic network. Such approaches, however, require that encryption keys for nodes, or groups of nodes, are known by the original message source.

Alternative methods for preserving privacy of communication patterns have also been proposed. To provide authentication along with source anonymity, Ahmad *et al.* describe using source pseudonyms with blind signatures [2]. This approach relies on a globally-trusted certificate authority, however, to sign pseudonymised certificates. Kate *et al.* [85] in the context of opportunistic networks, and Lu *et al.* [97] for VANETs, suggest using special trusted static nodes, which they respectively term “DTN Gateways” or “Roadside Units”. In each case, senders first pass their message to the trusted node, which is trusted to know their identity. Then this node forwards the message (along with other messages similarly collected) to the network. This assumes, however, the presence of these fully-trusted static nodes.

Solutions that can be implemented locally at nodes, without requiring encryption or globally-trusted third parties, have also been suggested. In the context of an opportunistic publish-subscribe application, Döra and Holczer [50] demonstrate a method to perturb user profiles, to avoid traffic analysis revealing a user’s specific interests. Or for more traditional opportunistic network routing, Radenkovic *et al.* [118] propose AdaptAnon, an anonymisation overlay on top of opportunistic network social network routing, where local heuristics are used to build on-demand “anonymisation paths” for messages through the opportunistic network.

Summarising, unless special care (such as onion routing) is taken to preserve anonymity of communication patterns, traffic analysis may reveal movement of messages throughout the network. We discuss the implications further in Chapter 4.2.

3.2.3 Location privacy

Users may not wish their physical locations to be public knowledge [16, 28, 40, 15, 6, 61]. Participating in an opportunistic network exposes users to location privacy threats. For example, Ristanovic *et al.* describe how location patterns can be recovered by analysing node encounters (such as those which would be detected through using an opportunistic network), if some malicious nodes are logging GPS locations [121]. But while techniques to preserve location privacy are well-studied in other contexts — such as protocols in sensor networks [83] and MANETs [56], or by cloaking or obfuscating specific locations when using services explicitly dedicated to publishing sensed locations [146, 48, 90, 7] — preserving location privacy is a relatively new research topic in networks such as opportunistic networks.

To address the location privacy threats, Lu *et al.* introduce ALAR, an opportunistic network routing protocol designed to be resistant to localisation attacks [98]. The technique is based on splitting each message into multiple segments, and then sending each segment encrypted via multiple paths. The decryption key is included with one of the segments, sidestepping the key distribution problems described in previous sections. This, however, degrades routing performance, because the message cannot be read until the receiver has received multiple segments — which is less likely than receiving a single message.

A second, alternative approach is taken in three similar opportunistic network routing protocols, all introduced by Zakhary *et al.*: HSLPO [152], SLPD [151], and LPAF [153]. Each protocol relies on forwarding messages in two phases: an initial obfuscation phase where only trusted social contacts are utilised for forwarding, followed by a second free-forwarding phase where a non-privacy-aware routing protocol may be used. The intention is that no untrusted node (i.e., not a trusted social contact) will receive the message until it has spread sufficiently-far from the original sender to provide this sender location privacy.

Location privacy is a relatively new research topic for opportunistic networks. Eavesdropping can, however, unless countermeasures are taken, allow inferences to be made about location or co-location. We return to this point in Chapter 4.2, and focus on examining users' attitudes to location privacy throughout Chapter 4.

3.2.4 Social graph privacy

The final type of potentially-private information that we consider is the social graph, where nodes are opportunistic network users and edges are social links. Such graphs are, as described in Chapter 2, useful for social network routing. The structure of a social network graph may in itself be private information [105, 100]. To demonstrate this, Narayanan and Shmatikov show that it can be possible to link two distinct social networks based on network topology alone, and that this may therefore allow deanonymisation of nodes by comparison of an "anonymised" social network to another available social network dataset [105].

Focusing on one aspect of social graph privacy — preserving the privacy of the probability that a given node will encounter a particular other node, which may correlated with social relationships — Hasan *et al.* introduce the 3PR routing protocol [69]. Nodes are organised into communities, and within communities messages are forwarded using epidemic-style forwarding. Between communities, privacy is preserved by exchanging community-level encounter probability information only — rather than an individual node's probability information — in an otherwise PRoPHET-style routing decision. 3PR relies on the assumption that nodes can be grouped into communities, however, and also does not consider social network routing, where social network information explicitly drives routing decisions.

Approaching the problem from the other direction, researchers have considered how to find social relations in opportunistic networking scenarios without compromising privacy. Guo *et al.* [65] introduce PSaD, a privacy-preserving scheme to disseminate content to social relations — where social relations are defined in this context as users with similar attributes. Similarly, Costantino *et al.* [44] describe the privacy vs accuracy trade-off in interest-casting, where the goal is to find "friends" — which they define as users with similar interests — in a privacy-preserving manner. Both schemes, however, define social contacts as users interested in receiving similar content. They do not consider general opportunistic network routing, and specifically social network routing — where pre-existing friendship links may be used to drive routing decisions.

As described in Chapter 2.2.3, potentially-sensitive social network information may be used within opportunistic network social network routing protocols to enable performant routing. Privacy issues are, however, typically not considered. Even for the one social network routing protocol that we are aware of where privacy-preserving schemes are discussed — the HiBOp scheme introduced by Boldrini *et al.* [25] — performance evaluation is conducted in the absence of the privacy-preserving features. We therefore highlight a gap in current opportunistic networking research: whether social network information can be used for social networking routing in a privacy-preserving manner, without diminishing network performance. We focus on examining this open research topic within Chapter 5.

3.3 Security attack defences

Malicious nodes within opportunistic networks may perform attacks [36], a sample of which we have previously enumerated in Chapter 2.3.2: black hole [49], Sybil [51], routing information falsification [30], and denial-of-service (DoS) via flooding [144]. While opportunistic networks have an inherent degree of resilience to some attacks [30], such as the black hole attack, they are not necessarily infallible; attacks have been demonstrated to impact network performance [37].

Defences to these attacks have been well-studied for always-connected networks, such as MANETs, but often do not generalise to networks that are expected to exhibit frequent (if not permanent) disconnectivity, such as opportunistic networks.[1] For example, Deng *et al.* describe a modification to the MANET AODV routing protocol to provide resilience to black hole attacks, by sending test data upon route establishment prior to using the route [49] — but opportunistic network routing protocols do not establish fixed routes, and so cannot use this method. Other examples of protocols that similarly depend on always-connected networks, and thus are not suited for the generalised case of typically-disconnected opportunistic networks, include the MANET secure route discovery protocol proposed by Papadimitratos and Haas [109] (which again requires fixed routes), and, for flood-

[1] In special cases, an opportunistic network may exhibit periods of high connectivity — for example, if a crowd of people are for a short time in a densely-packed small area, such as a full stadium. But in the general case, opportunistic networks are expected to experience frequent disconnectivity, and therefore opportunistic network protocols are necessarily designed to function in disconnected scenarios.

ing attacks, the MANET Flooding Attack Prevention scheme introduced by Yi *et al.* [147] (which relies on deprioritising route establishment during busy periods and deleting existing routes that are being used to flood). Outside of MANETs, Yu *et al.* describe SybilGuard [149], a method to detect the Sybil attack via social network analysis to detect abnormality — but which again relies on a connected network, and so is not suitable for opportunistic networks.

Other defence schemes may not be inherently inapplicable for opportunistic networks, but in practice are not suitable. Kim and Helmy describe CATCH, a cross-layer MANET framework for traceback [87], i.e., a method to determine the source of malicious traffic in order to allow it to be blocked. The detection mechanisms proposed rely on nodes near to the attacker being able to detect abnormal traffic — an assumption that is likely to hold for MANETs, where nodes are typically in close proximity at all times, but not for opportunistic networks, where encounters are more sporadic. Some other examples include the MANET flow-based flooding detection proposed by Guo *et al.* [66] (since traffic flows are typically bursty for opportunistic networks, i.e., only during encounters), and the Sybil detection proposed by Piro *et al.* [117] (since the method relies on repeatedly overhearing traffic from other nodes).

More broadly, some proposed security schemes — even those proposed in the context of opportunistic networks — rely on assumptions that may not hold in opportunistic networks. For example, in Chapter 3.2, we noted that the assumption of a PKI or globally-trusted third party is potentially unrealistic for opportunistic networks. Li *et al.* propose a scheme to detect and mitigate opportunistic network black hole attacks, using signed encounter tickets to provide evidence of encounters [94], but the scheme relies on this assumption. A second such strong assumption, as previously discussed, is requiring the existence of globally-trusted nodes. Examples of schemes making this assumption include SPRING introduced by Lu *et al.* [97] and FBIDM by Chuah *et al.* [38]; these schemes allow detection of black hole attacks by requiring the existence of static globally-trusted VANET nodes (“Roadside Units”) or mobile globally-trusted opportunistic nodes (“ferries”) respectively.

We see that there is a need to develop schemes to defend against security threats to opportunistic networks, without requiring such assumptions. This is an ongoing research area, although some such schemes have been proposed.

For example, consider black hole attacks. Uddin *et al.* introduce SPREAD [140], a scheme where opportunistic network nodes increase the number of copies sent upon locally detecting evidence that messages are not being delivered by other nodes within the network. Al-Hinai *et al.* describe TB-SnW (Trust-based Spray-and-Wait) [3], an opportunistic network routing protocol designed to be resilient to black hole attacks by nodes locally maintaining "trust" values for other encountered nodes — providing greater trust to nodes which have historically participated in exchanging messages. Zakhary and Radenkovic show that erasure coding — a type of networking coding where the original sender splits messages into multiple parts, which are sent independently, and not all of which need arrive at the destination for successful message delivery — can also help to mitigate black hole attacks [150].

More generally, Trifunovic *et al.* consider trust in opportunistic networks — both in the context of detecting Sybil nodes [139] and detecting unwanted spam messages [138]. In [139], Sybil nodes are detected by a scheme where nodes exchange lists of their friends — trusted social contacts — during encounters. Each node is then able to locally construct a view of the whole social graph, and assigns greater trust to closer nodes (e.g., friends-of-friends). This scheme does not address privacy, however; we noted in Chapter 2.3.1 that the social graph is potentially sensitive, and so it may be undesirable to share friendship information publicly. In [138], trust values are assigned to all other encountered nodes in the network. Users are required to explicitly classify messages received as either spam or legitimate content — although classifications are shared between nodes, to avoid every node having to classify every message. This scheme is therefore not usable for message delivery in networks where message payload is considered sensitive: intermediate forwarding nodes cannot classify a message as spam or legitimate without reading the plaintext message contents.

Summarising, Table 3.2 categorises this sample of defences to the security threats noted in Chapter 2.3.2, dividing research between MANETs and opportunistic networks. We note from this a research gap: automatic detection and mitigation of flooding attacks in opportunistic networks has not been addressed, except by schemes requiring strong assumptions such as a globally-trusted third party. The

	MANET defence	**Opportunistic network defence**
Black hole	[49]	[3, 38, 94, 97, 140, 150]
Sybil	[117, 149]	[139]
Routing information falsification	[109]	N/A (no routes to falsify)
Flooding	[66, 87, 147]	-

Table 3.2: Research to address security threats in MANETs and opportunistic networks.

closest related work is the PRED queueing policy introduced by Lee *et al.* [93]. PRED is a buffer queueing policy for probabilistic opportunistic network routing protocols (e.g., PRoPHET). PRED attempts to deprioritise storage of flood messages, to ensure that legitimate messages displace flood messages in full buffers. While this may help to avoid legitimate messages being dropped due to full buffers, such a queueing scheme does not fully mitigate the flooding attack: other costs to nodes are still incurred by the attack, such as energy loss.

3.4 Summary

In this chapter, we have outlined the current state-of-the-art opportunistic networking security and privacy research. We have noted the following points, and three open questions:

- Research has focused on routing protocol performance in opportunistic networks, primarily evaluated through simulation. An open question is whether performance may be impacted by users reducing their participation in the network due to privacy concerns.

- Potentially-sensitive social network information may be used within routing protocols, to enable performant routing. An open question is whether the social network information may be used in a privacy-preserving manner, while maintaining routing performance.

- Existing proposed schemes to mitigate security threats often rely on strong assumptions which may not hold across many opportunistic network deployments — for example, the presence of a globally-trusted third party. An open question is how to mitigate threats such as denial-of-service flooding attacks while relaxing these assumptions.

We examine each of the three open questions noted above in turn, one each per chapter in Chapters 4–6. In the next chapter, we therefore consider the first of these open questions: if users reduce their participation in opportunistic networks due to their privacy concerns, then might this impact the performance of the network?

Chapter 4

Exploring the performance impact of users’ privacy preferences

4.1 Introduction

We have noted in Chapter 2.3.1 that participating in an opportunistic network can expose users to privacy threats, and in Chapter 3.1 that users may become less wiling to forward messages on behalf of other users because exposure to the privacy threats is minimised during times when the opportunistic network application is not running. We have also noted the research gap, where opportunistic network performance simulations have not considered how to measure and model such privacy-preserving user behaviour, and its corresponding impact on network performance.

To simulate network performance taking account of this privacy-preserving behaviour, we require a model of the users’ privacy-preserving behaviour. By analogy to user mobility models, as we have seen in Chapter 3.1.2, one way to obtain such a model is to create a synthetic model generated according to mathematical properties. Synthetic models, as we have seen, are useful for simulation since they are easy to generate, but may have limitations in how accurately they represent real user behaviour.

An alternative is to build an empirical privacy model, based on measurements of users' behaviour, to then apply to opportunistic network performance simulations. But a new question then arises: if we perform a user study using a simulated application, would the privacy preferences recorded match that when using a real application? Might this affect the empirical privacy models obtained from such an experiment, and therefore the opportunistic network performance results?

The contributions of this chapter are:

- We apply a synthetic privacy model to opportunistic network performance simulations, and demonstrate that a large performance impact can result.
- We construct an empirical privacy model based on actual measurements of people. Applying the empirical privacy model to opportunistic network simulations, we demonstrate that it may be possible to use simulated applications as a substitute for real applications when measuring privacy preferences. We also demonstrate that users' privacy preferences can have a large impact on network performance.

4.2 Attack model

In Chapter 2.3.1, we discussed examples of privacy threats to which opportunistic network users may be exposed. We consider first how an attacker could perform attacks.

We consider an attacker with certain, limited capabilities. From the attacker models enumerated in [92] against opportunistic networks, our interest is in the *local eavesdropper* (an attacker who can eavesdrop in the vicinity of a user), and the *partial eavesdropper* (an attacker who can place receivers in a number of hotspots and intercept traffic in the vicinity). We agree that a *global eavesdropper* is not a practical attack model in an opportunistic network — by the very nature of such a network, nodes are distributed over a very large area, and traffic is not routed through any central hub. Therefore we do not consider attacks which would require global knowledge, such as an attacker studying overall network traffic patterns. We enumerate attacks based on intercepting some number of messages.

We choose to employ *attack trees*, as introduced by Schneier [122]. An attack tree is a type of *and-or tree*, used to enumerate attacks against a system. The root node of the tree is the overall attack goal, while nodes within the tree are subgoals. The children of a particular node are the steps required to achieve that node's subgoal. By constructing such a tree from the root node (overall goal) downwards, we now enumerate a structured threat analysis for attacks against an opportunistic network using social network routing. We select the following goals as "low-hanging fruit" for an attacker intent on compromising privacy of opportunistic network users.

Goal 1: Discover structural information about the social network graph. (Threat to social graph privacy.)

1. Learn whether a friendship link exists (or does not exist) between two users. OR
    (a) Discover communication (or lack of) between the users. OR
        i. Eavesdrop a message as it is forwarded user-to-user, from source to final destination (or any intermediary). OR
            A. In social network routing protocols such as SRSN Routing (SRSNR), a message traced along such a path reveals social network links (or lack of) — because messages are forwarded if and only friendship links exist. Friendship links are the path traversed by the message.
        ii. Extract source/destination from an intercepted message to an intermediary.
    (b) Extract friendship links from an intercepted message to an intermediary.
2. Learn how many friendship links a particular user has.
    (a) Extract friendship links from an intercepted message to an intermediary.

Goal 2: Discover whether two individuals have been in proximity within a certain timeframe. (Threat to location privacy.)

1. Follow one or both individuals for the time in question. OR
2. Infer proximity by sending a specially crafted message, and making inferences based on where the message is observed within the network. OR
    (a) Example: has Alice from New York recently met Bob from Los Angeles? To find out, an attacker Mallory in New York can inject a message addressed for colluding attacker Trudy in Los Angeles into the system, with Alice and Bob only as requested intermediaries. If Trudy receives Mallory's message, Mallory and Trudy have learned that Alice and Bob have met within the lifetime of this malicious message.
3. Infer proximity by noting that messages are not forwarded twice. OR
    (a) Example: if a message is not forwarded to a node known to be a requested intermediary, the message must already have been forwarded earlier. An attacker can infer that the nodes were in proximity before this time. This is a passive version of 2, not requiring message injection.
4. Wait in a common place and listen for message traffic. Message exchange, or message headers, may reveal the colocation of individuals to an attacker.

Goal 3: De-anonymise a social network to discover the presence of individuals within the network. (Threat to anonymity.)

1. Follow individuals, and tie their network identifiers to their actual identities. OR
2. Infer identities from known portions of the social network.
    (a) Example: if five people are known to be mutual friends, and four are deanonymised with a fifth mysterious node, an attacker can infer that this unknown node is the last member of the clique.

For each of these privacy threats, an opportunistic network user can mitigate the risk by reducing their participation in the network. How would this privacy-preserving user behaviour affect opportunistic network performance?

4.3 Performance evaluation: synthetic privacy model

We first consider simulation privacy-preserving user behaviour using a *synthetic* privacy model. As previously noted in Chapter 3.1.2, synthetic models are already widely used in opportunistic network research in the context of mobility. In this section, we extend the SRSN Routing (SRSNR) protocol as discussed in Chapter 2.2.3 with a simple synthetic privacy model.

In SRSNR, messages are forwarded between the social contacts ("friends") of the original message sender, during encounters when they are in physical proximity. We introduce a synthetic privacy model to this by assuming that nodes do not always participate in exchanging messages during each encounter, but would rather exchange messages during an encounter only with some fixed probability; this probability is a parameter of the synthetic privacy model. For example, with the probability parameter of 40%, in a given encounter two nodes would perform SRSNR-style message exchange with one another with probability 40%, otherwise no messages would be exchanged. Similarly, if the probability parameter was set to 0%, this would imply that nodes cared so much about their privacy that they refused to share data with any other nodes. The baseline behaviour of SRSNR in the absence of a privacy model corresponds to a probability parameter of 100%.

To evaluate the performance of opportunistic networks employing this synthetic privacy model with varying probability parameter, we use trace-driven simulation.

4.3.1 Datasets

We use three real-world datasets to evaluate our routing schemes. While there are a variety of available datasets including encounter and social-network information, the three datasets used were chosen for their different scale and structure. All datasets are publicly available.

1. The *SASSY* dataset [21]. In this dataset collected at St Andrews, 25 participants were equipped with 802.15.4 Tmote Invent sensors, and tracker for 79

days. We use an augmented version of the trace, as detailed in Appendix A.1.1, resulting in a dense trace of encounters between participants. Social network information was obtained from Facebook friendships.

2. The *Reality Mining* dataset [53]. In this well-known dataset collected at MIT, 97 university members carried mobile phones during their daily lives over the course of an academic year. The phones recorded the results of periodic Bluetooth scans. We define Bluetooth encounters between participant devices as opportunities for message exchange in an opportunistic network, and use mobile phone address book contacts to determine social network information.

3. The *LocShare* dataset [13]. As we will see later in this chapter, the *LocShare* dataset was gathered for the primary purpose of obtaining an empirical privacy model. During the user study, however, we also collected location and social network information, and therefore have a third dataset. We collected the locations of 80 participants during four one-week runs of 20 participants. By defining an encounter as occurring when two participants are within 10 metres — selected as this is the approximate average Bluetooth range — we obtain a trace of encounters between participants. As for the *SASSY* dataset, social network information was obtained from Facebook friendships.

Detailed descriptions of the datasets — including a comparison of their diverse scale and structure — may be found in Appendix A.

4.3.2 Simulation parameters

We use the following set of parameters for the simulations:

- 100 runs per data point. Multiple runs were performed in order to allow confidence intervals to be determined; 100 runs were chosen as the highest order-of-magnitude to perform while remaining computationally tractable.
- Unicast messages, sent from a random sender node to a randomly-chosen social network neighbour (*friend*) of this sender. Note that although messages

are unicast (destined for one particular recipient), the messages may follow multiple paths through the network in order to reach that destination. Note also that this is a random traffic pattern, not based on real data; simulating a real traffic pattern is considered out of scope for this thesis (see Chapter 7.3).

- 100 messages per run.
- A message time-to-live (TTL) of one day.
- One week per simulation.[1]
- Infinite buffers and infinitely-fast transmission.[2]

We consider probability parameters for the synthetic model from 0% to 100%, in steps of 20%.

4.3.3 Results

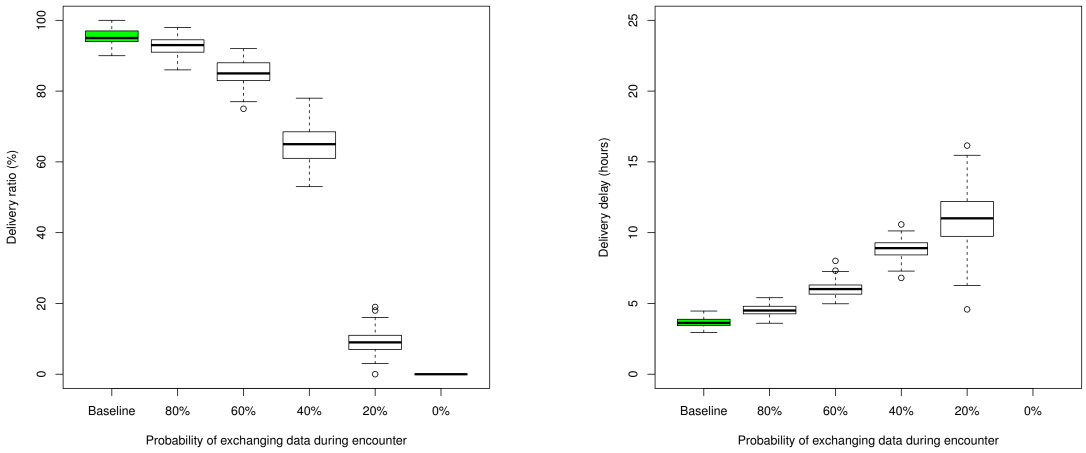
Figures 4.1–4.3 show the performance of the network, as measured by two commonly-used metrics [75]:

- *Delivery ratio:* proportion of delivered messages, out of the total number of unique messages generated.
- *Delivery delay:* time taken for a message to first reach its destination.

Figure 4.1(a), Figure 4.2(a) and Figure 4.3(a) show that for each dataset, the network performance as measured by delivery ratio strongly depends on the chosen probability parameter. There is wide variation in performance, with performance declining with decreasing probability of message exchange during each encounter.

[1] For *LocShare*, there are four one-week parts to the dataset; we therefore simulate 25 runs with each of the four one-week parts to make up the 100 runs for each datapoint. For *Reality Mining*, we pick a random one-week interval for each of the 100 runs — but we select only one-week intervals where there are sufficient numbers of nodes present for non-trivial routing to be possible. For *SASSY*, the augmented version of the trace is for a 30-day segment.

[2] Our goal is to investigate the performance impact of privacy, so we do not set arbitrary constraints on buffer size or transmission rate, as these may confound the results.



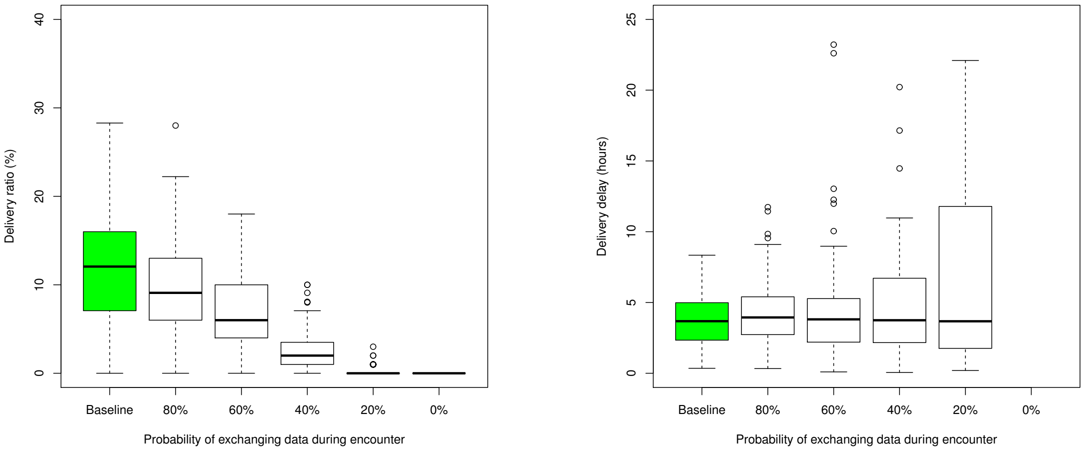
(a) Delivery ratio.

(b) Delivery delay.

Figure 4.1: *SASSY* dataset: Delivery ratio and delay under the synthetic privacy model. Delivery ratio and delay vary widely based on the chosen probability parameter.

For the *SASSY* dataset, Figure 4.1(b) shows a similar trend for delivery delay: performance decreases (i.e., delivery delay increases) with decreasing probability parameter. For the *Reality Mining* and *LocShare* datasets, however, Figure 4.2(b) and Figure 4.3(b) show that delivery delay does not significantly change. In these sparser datasets, any impact on delivery delay of changing the probability parameter is lost in the noise. We also note a paradoxical result for the *LocShare* dataset: median delivery delay is low for the 20% probability parameter. This is a survivor effect: delivery delay calculations ignore messages which are not successfully delivered, and so the successfully-delivered messages necessarily take shorter paths to their destination, since message exchange is unlikely during each encounter.

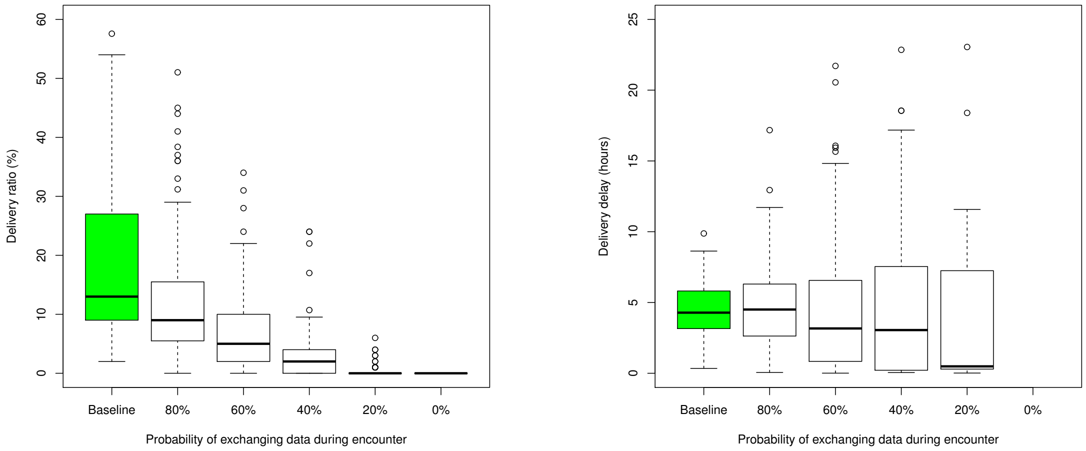
While there is wide variation in network performance based on the probability parameter, a limitation of the synthetic model is that there is no obvious way to determine which probability parameter to use. Delivery performance depends on the extent to which users reduce their participation in the network to preserve their privacy. We therefore next consider how to construct an empirical privacy model.



(a) Delivery ratio.

(b) Delivery delay.

Figure 4.2: *Reality Mining* dataset: Delivery ratio and delay under the synthetic privacy model. Similarly to the *SASSY dataset* (Figure 4.1(a)), there is wide difference in delivery ratio based on the chosen probability parameter. Delivery delay does not significantly change, however; this may be an artifact of the sparse encounter trace, since any successfully-delivered message will necessarily take approximately the same, small, number of hops.



(a) Delivery ratio.

(b) Delivery delay.

Figure 4.3: *LocShare* dataset: Delivery ratio and delay under the synthetic privacy model. As for Figure 4.1(a) and Figure 4.2(a), delivery ratio varies widely based on the chosen probability parameter. Paradoxically, delivery delay is reduced when data are less likely to be exchanged during each encounter; successfully-delivered messages necessarily take shorter paths to their destination, since message exchange is unlikely during each encounter.

4.4 Performance evaluation: empirical privacy model

4.4.1 Methodology

We performed a user study to investigate the location-sharing privacy preferences of 80 users of the popular Facebook social networking site. By analogy to existing location-sharing applications, we introduce the assumption that privacy choices for location-sharing behaviour when broadcasting locations via Facebook will not be dissimilar to those for an opportunistic network participant, i.e., that users would be willing to participate in an opportunistic network at times that they were willing to share their location. We can therefore develop an empirical privacy model for opportunistic network usage based on the location-sharing privacy preferences measured in this study.

Participants in the LocShare experiment carried a location-sensing mobile phone for one week of their day-to-day lives. Due to resource constraints — we had 20 mobile phones available, but 80 participants — the user study took place over four one-week runs, each with 20 participants. Two runs each were performed in a UK town (St Andrews) and a UK city (London). Participants were selected from undergraduate students in both locations. We also selected only students not studying Computer Science, so that they would not be known by us, and who self-reported as daily Facebook users.

Following the Experience Sampling Method (ESM) [46, 41], participants were prompted *in situ* to answer questions relating to their location privacy preferences. Each participant was prompted up to 20 times per day to choose how widely their current location could be published on Facebook — to *everyone*, to some or all of their Facebook social contacts (*"friends"*), or to *nobody* at all.

We asked participants to answer as many questions as possible, but we gave the option of ignoring questions in order to avoid false responses. A total of 7,706 prompts were sent, which resulted in 4,232 replies (a 54.8% response rate). Individual participant response rates ranged from 15.7% to 91.4%. Compensation was not linked to response rate, which may be a factor in the response rate being lower than in some other mobile ESM studies [40, 60]. The average response rate was 7.6

Category	Proportion in category	Location sharing choice		
		Nobody	Friends	Everyone
Open	18% (7/40)	5.2%	7.7%	87.1%
Social	45% (18/40)	9.7%	80.5%	9.7%
Closed	28% (11/40)	72.1%	20.3%	7.6%
Variable	10% (4/40)	33.3%	29.6%	37.2%

Table 4.1: Real group ($n = 40$). Rows may not add up to 100% due to rounding.

replies per participant per day.

In addition to measuring how widely participants were willing to share their current locations, we had a secondary goal of investigating differences in privacy behaviour between users of a real and a simulated application. During each run, participants were therefore divided into two groups. Half of the participants (10/20) were assigned at random to the *real* group, and half to the *simulation* group. Participants in the real group had their locations published to Facebook, visible to their social contacts according to their chosen preferences. Participants in the simulation group were able to see on Facebook the information which would have been published, but this was not disclosed to any of their social contacts.

We informed participants to which group they were assigned at the beginning of each run. While we considered a blind experiment, where participants would be unaware of their group, we explicitly told participants their group in order to better match the scenario of the simulation group to prior simulation-only studies, such as [40], where *all* participants used a simulated system, and were aware of this fact throughout.

4.4.2 Model

We use the privacy preferences reported by real users during the user study to obtain an empirical privacy model. To study the differences between simulated and real applications, we construct separate models based on responses in the real and simulation groups.

Category	Proportion in category	Location sharing choice		
		Nobody	Friends	Everyone
Open	20% (8/40)	10.0%	3.9%	86.1%
Social	53% (21/40)	5.8%	75.4%	18.8%
Closed	18% (7/40)	67.0%	20.5%	12.6%
Variable	10% (4/40)	32.2%	36.4%	31.4%

Table 4.2: Simulation group ($n = 40$). Rows may not add up to 100% due to rounding.

As is common in other privacy models [143], we segment the participants of each group into categories according to their privacy behaviour, i.e., their responses to the prompted questions (see Tables 4.1–4.2). We define four categories:

- *Open*: Participants usually shared their location publicly with everyone, in over 50% of responses.
- *Social*: Participants usually shared their location with some or all of their Facebook friends, in over 50% of responses.
- *Closed*: Participants usually did not share their location to anybody at all, in over 50% of responses.
- *Variable*: Participants did not have consistent location-sharing behaviour. They would sometimes share with nobody, with friends, and with everyone.

Note that, to a certain extent, all of our users had "variable" behaviour in that they did not act consistently at every location.[3] Thus for the *Open*, *Social* and *Closed* groups, we consider a participant to be a member of this group if their behaviour is consistent with this group in over 50% of their sharing activity, i.e., their responses to questions.

For each of the four categories, we calculate the mean of the users' location sharing choice proportions (*nobody*, *friends* or *everyone*) by user, in order to obtain Tables 4.1–

[3] Ben Abdesslem *et al.* explore the implications of this further in a position paper [12], arguing that contextual information should be considered when examining users' privacy preferences.

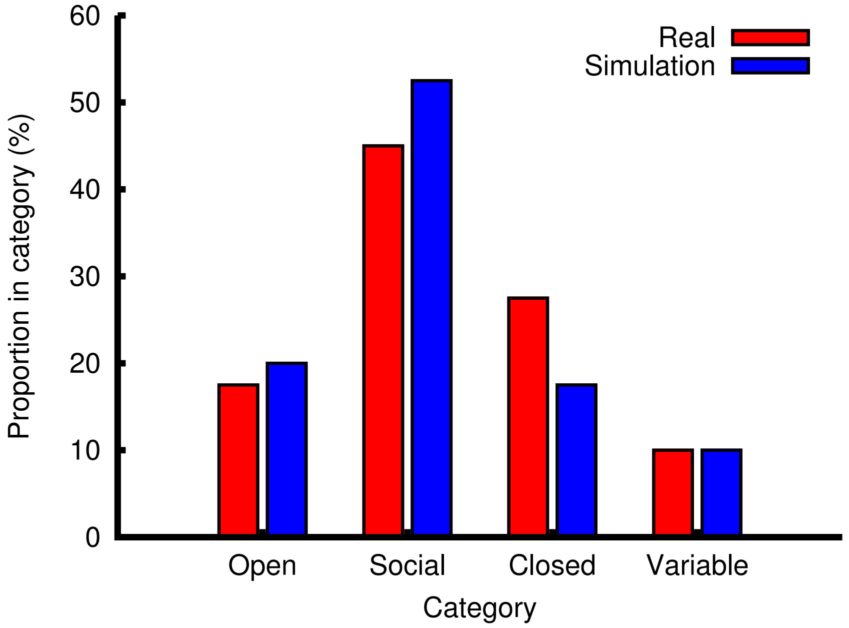


Figure 4.4: Proportion of participants in each category of the privacy model.

4.2. Using these statistics, we can construct *privacy models* for users' location sharing preferences, for each of the two groups of users. These privacy models are dataset-independent, and so may be applied to a variety of datasets for opportunistic network routing simulations. While it may be possible to construct more sophisticated models [137], our purpose here is to examine the differences in models between real and simulated applications, and so we limit ourselves to simple statistical models for now.

Figure 4.4 shows a visualisation of the relative number of participants in each of the four categories, for the real and simulation groups. The distribution of participants across the categories is generally similar between the real and simulation groups. As perhaps may be expected, most social network users were indeed social, and willing to share with some or all of their Facebook friends. Smaller proportions were either more open or closed than this, or acted in a variable fashion. Note, however, that participants in the real group seemed more privacy-concerned than those in the simulation group: more participants in the real group were assigned to the privacy-concerned *Closed* category, and fewer to the less-privacy-concerned *Open* and *Social* categories, than in the simulation group. This might also be expected; participants might have taken more care about sharing their information if they knew that information was actually being shared on Facebook, and so acted in a more privacy-conscious, and perhaps more realistic, fashion.

We can now apply these privacy models for the two groups to opportunistic network simulations, in order to examine to what extent these differences in privacy behaviour affect routing performance.

4.4.3 Datasets and simulation parameters

To study the impact of simulated and real social networking applications on opportunistic network routing, we perform trace-driven simulations with the empirical privacy models, using the same datasets and simulation parameters as in Section 4.3.

At the start of each simulation run, each node (i.e., simulated participant) is allocated to one of the categories (*open, social, closed, variable*) for the duration of the run. We perform simulation for three methods of category allocation:

- *Central nodes closed*: Similarly to [19], we make use of the finding that the altruism (i.e., willingness to participate in message forwarding) of high-degree nodes is most important for network performance [145], to highlight any performance impact from the privacy models in our simulations. We rank nodes from highest to lowest degree-centrality in the encounter graph, and assign nodes in order to the *closed*, *social*, *variable* and *open* categories — with the number of nodes in each category according to the proportional size of the category in the privacy model. Higher-degree nodes are therefore less likely to forward messages, since they are assigned to the less open categories — which is analogous to behaving less altruistically — and so any performance impact due to the privacy model is maximised.
- *Central nodes open*: As a baseline, we perform simulations with the opposite allocation strategy to *central nodes open*. The nodes are ranked from *lowest to highest* degree-centrality (the reverse ordering to *Central nodes closed*), and then allocated to categories as before. The lowest-degree nodes are then less likely to forward messages, while the highest-degree nodes are more likely

to do so.

- *Random category allocation*: As a further baseline, nodes are assigned randomly to categories, with allocation probability proportional to the size of each category. Centrality is not considered in the allocation process.

Algorithm 1 shows pseudocode for these three category allocation methods.

Algorithm 1 Nodes' category allocation

```
if allocation_scheme == 'random' then
    for all node in nodes do
        node_category ← random category (weighted by category size)
else
    ordered_nodes_stack ← []
    if allocation_scheme == 'central nodes closed' then
        ordered_nodes_stack ← [nodes ordered by descending centrality]
    else if allocation_scheme == 'central nodes open' then
        ordered_nodes_stack ← [nodes ordered by ascending centrality]
    for all category in categories do
        wanted_category_size[category] ←
        num_nodes * proportional_size[category]
    while node ← pop ordered_nodes do
        for all category in [closed, social, variable, open] do
            if num_in_category[category] < wanted_category_size[category]
            && node_category is unassigned then
                node_category ← category
```

We define two modes of privacy behaviour to apply the privacy models for the real and simulation groups to our opportunistic network simulations. While it is possible to think of many more behaviours, we believe that two modes are sufficient for investigating the impact of privacy. Previous work has also demonstrated that a constrained number of privacy choices is a usable compromise for privacy policies for ubiquitous computing environments [84]. Our chosen modes are:

- *Friendly (F)*: Nodes are modelled as being willing to share with their social network friends. If the overall privacy choice is *everyone*, then the nodes behave as in the default case; if *nobody*, then messages are not exchanged; if *friends*, then as the default case only if the two nodes involved in this encounter are friends (otherwise messages are not exchanged).

- *PubPriv (PP)*: Nodes are modelled as either being fully public (no privacy concerns), or fully private (any privacy concerns result in disregarding the encounter) — with nothing in-between. If the overall privacy behaviour during an encounter is *everyone*, then messages are exchanged as in the default case. Otherwise (i.e., if the overall privacy behaviour is *friends* or *nobody*), messages are not exchanged.

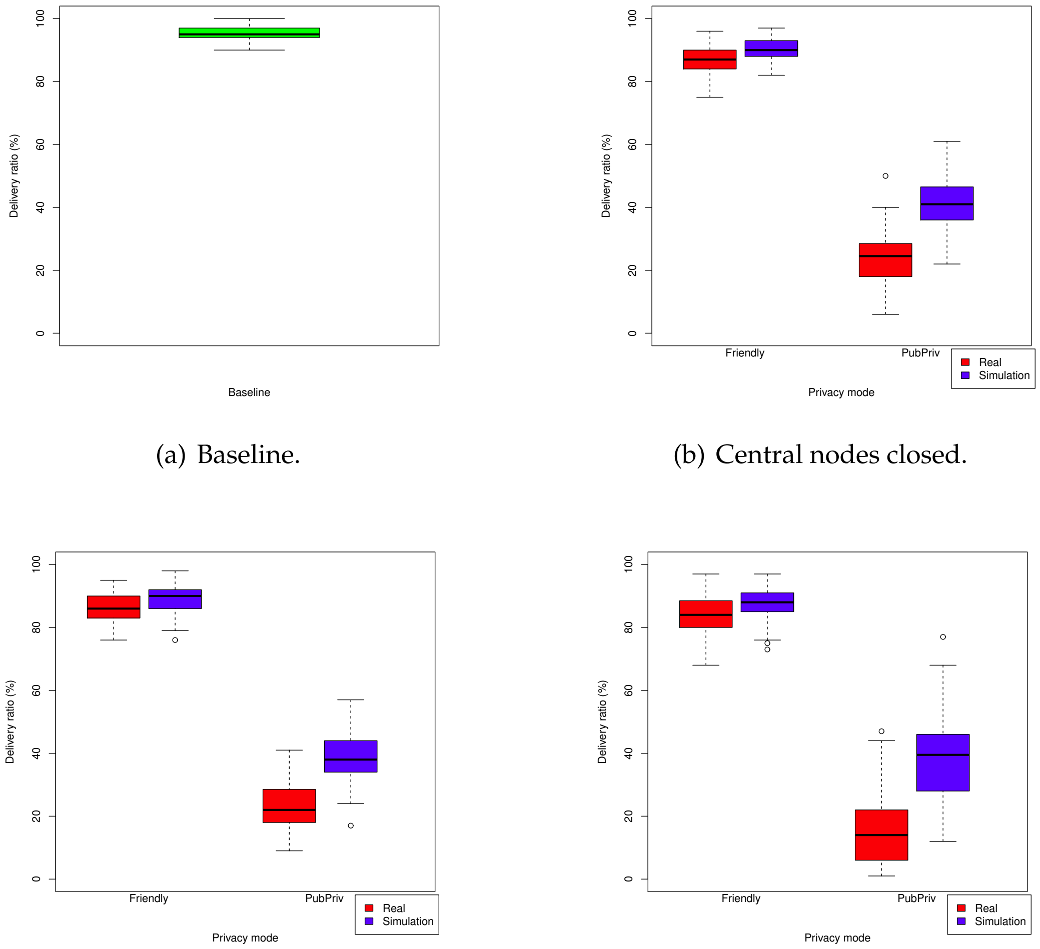
During each encounter between a pair of nodes, each of the two nodes randomly picks a privacy behaviour of $\{nobody, friends, everyone\}$, weighted according to the location-sharing proportions associated with that node's category. Messages are then exchanged depending on the chosen privacy behaviours for that encounter. The overriding choice is the more restrictive of the two nodes' privacy behaviours. For example, if one node picks *nobody* and the other picks *everyone*, then the overall choice is the more restrictive *nobody*.

Algorithm 2 shows pseudocode for the logic behind choosing whether messages are exchanged between nodes during each encounter, for each of the privacy modes.

4.4.4 Results

Figures 4.5–4.10 show the performance simulation results. Delivery ratios, as would be expected, are consistently lower in each case for the simulations using the privacy modes (Friendly or PubPriv) compared to the SRSNR baseline.

For the *SASSY* dataset, Figure 4.5 shows that delivery ratios are generally lower for the real group than the simulation group — significantly so for the PubPriv mode. Additionally, delivery ratios under the PubPriv mode are lower than the Friendly mode; the performance difference between the two privacy modes are significantly greater than differences between the real and simulation groups. Similar performance trends appear for delivery delays, as shown in Figure 4.6: lower performance (i.e., higher delivery delay) for the real group than the simula-



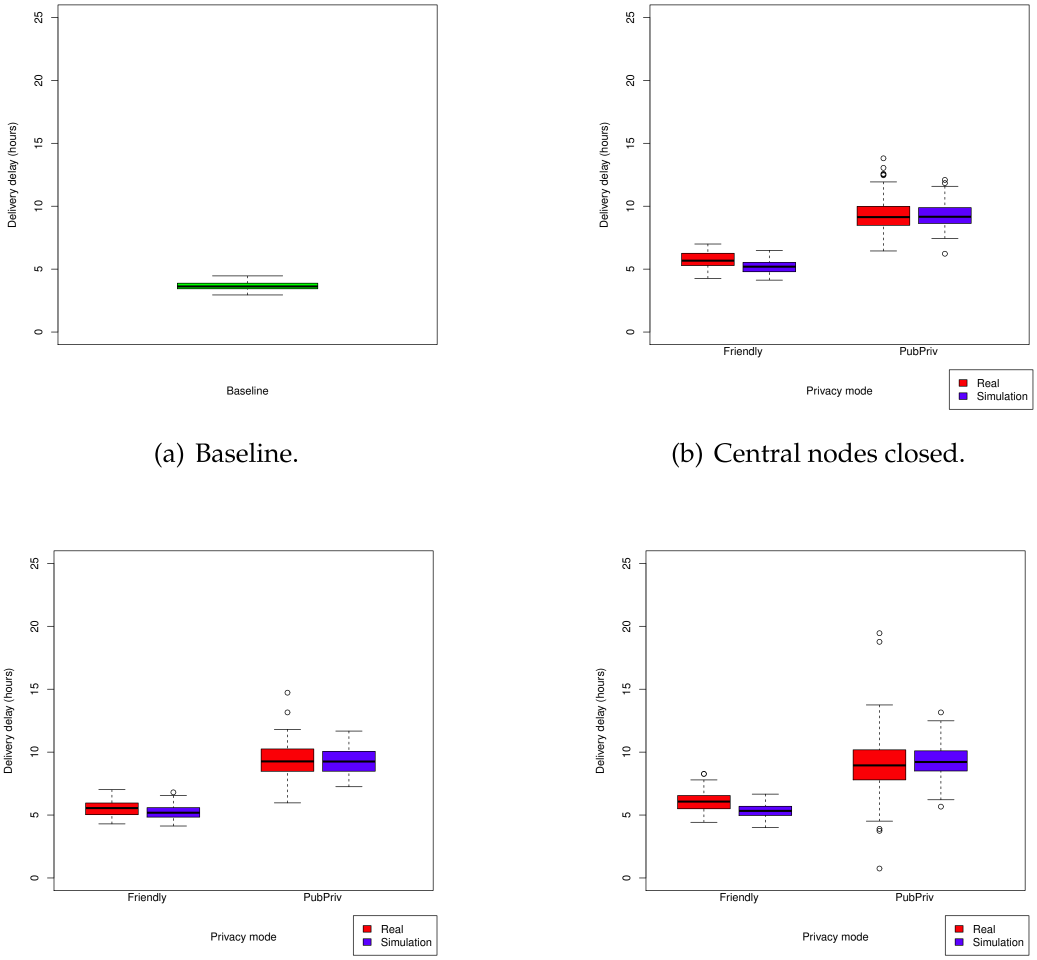
(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.5: *SASSY* dataset: Delivery ratio baseline, and delivery ratios for the real and simulation groups under two privacy models and three allocation methods. Delivery ratios under the two privacy modes are lower than for the Baseline, as expected. Delivery ratios are generally lower for the real group than the simulation group, and lower for the PubPriv mode than the Friendly mode; the differences between the privacy modes are significantly greater than the differences between the groups. In this densely-connected dataset, each allocation method's results are similar.



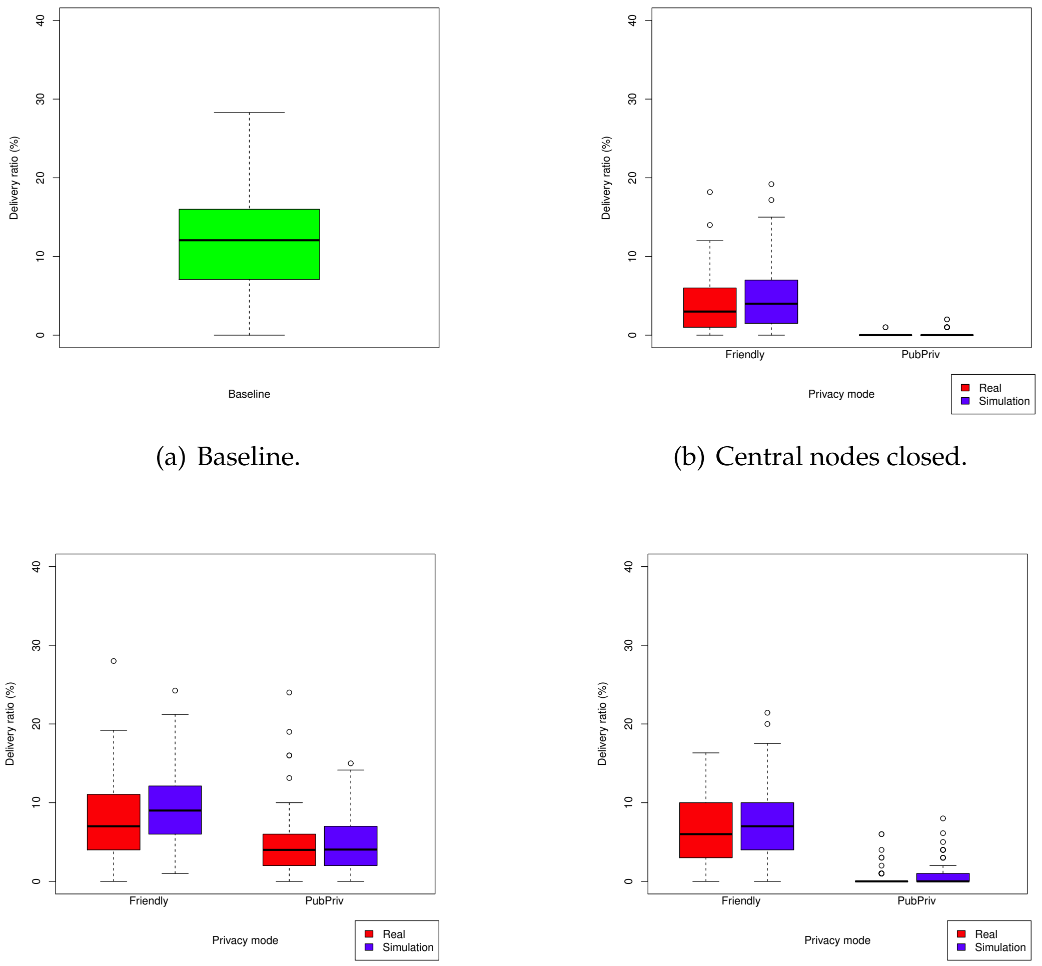
(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.6: *SASSY* dataset: Delivery delay baseline, and delivery delays for the real and simulation groups under two privacy models and three allocation methods. Performance trends are similar to Figure 4.5: lower performance (i.e., higher delivery delay) for the real group than the simulation group, and lower for the PubPriv mode than the Friendly mode, with differences greater between modes than between groups.



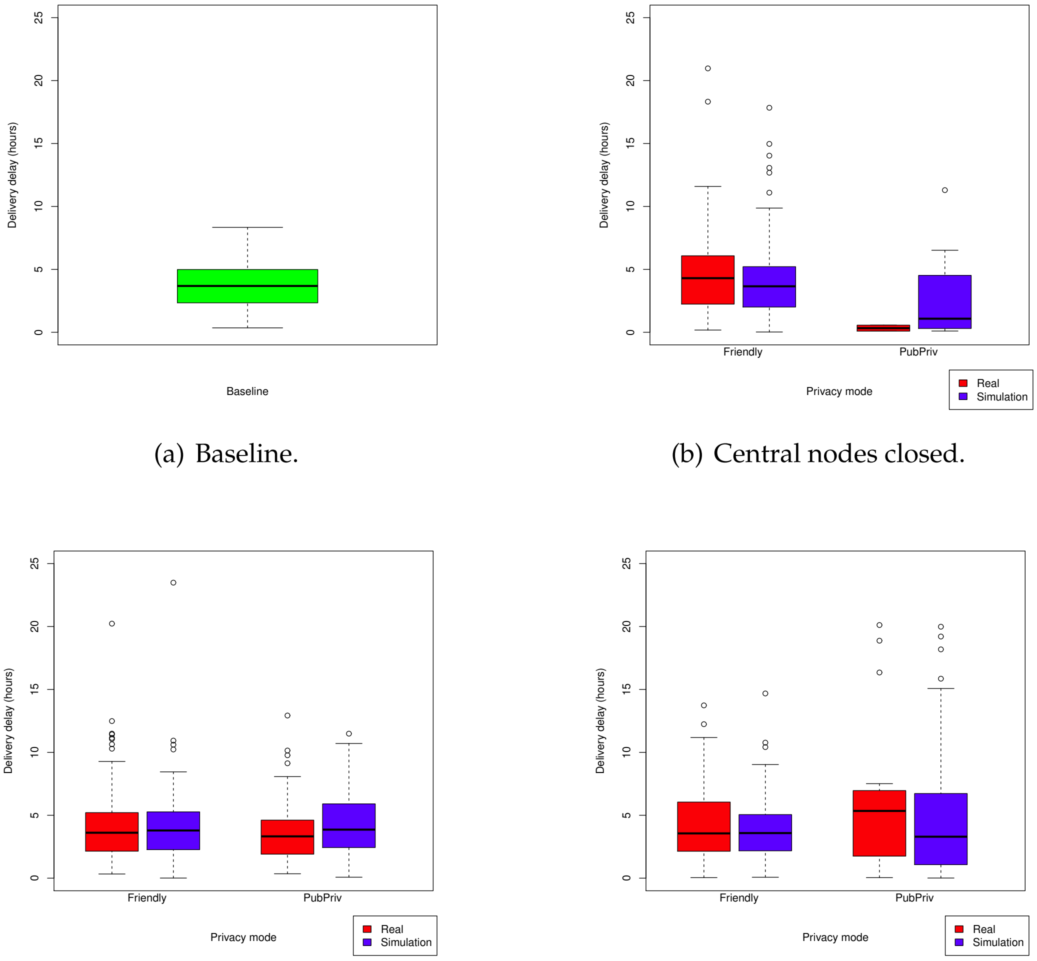
(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.7: *Reality Mining* dataset: Delivery ratio baseline, and delivery ratios for the real and simulation groups under two privacy models and three allocation methods. Delivery ratios are low for the Friendly privacy mode, and very low for the PubPriv mode — falling to zero when magnifying differences under the central nodes closed allocation scheme. Delivery ratios are similar for the real and simulation groups in each case.



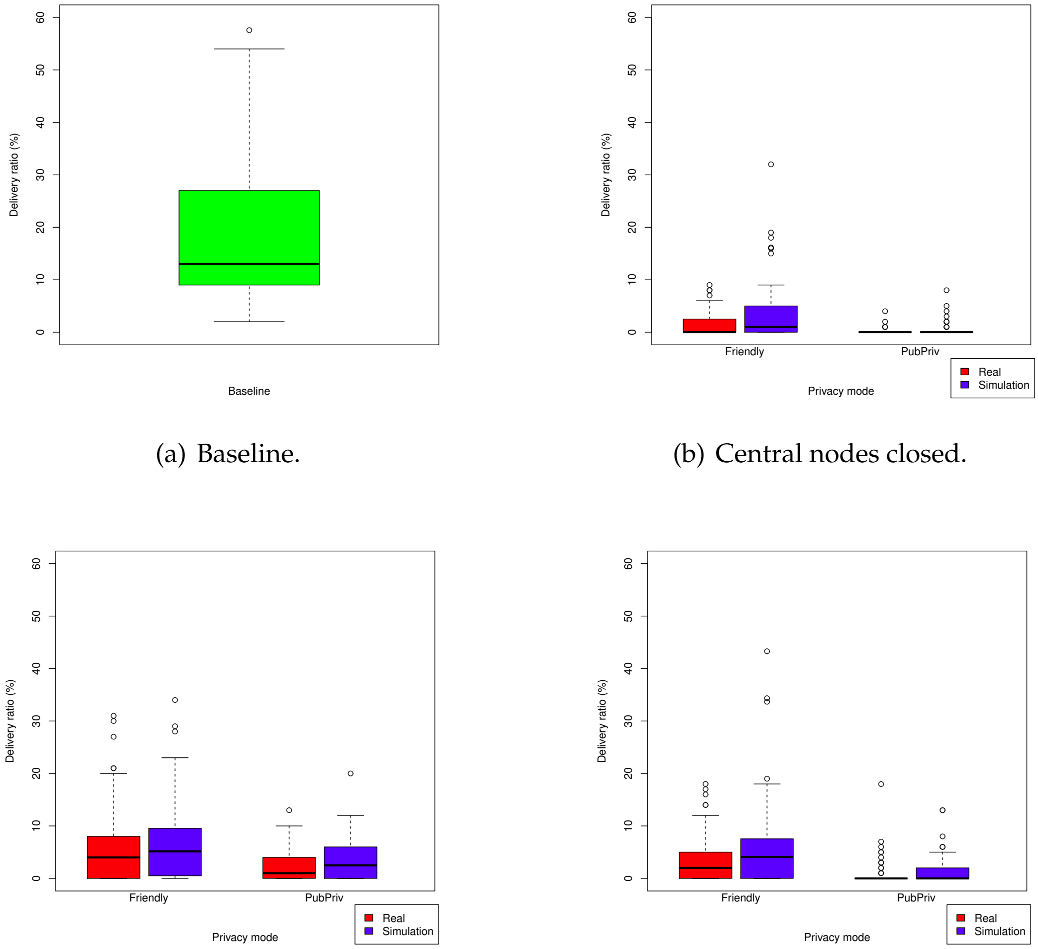
(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.8: *Reality Mining* dataset: Delivery delay baseline, and delivery delays for the real and simulation groups under two privacy models and three allocation methods. There are no significant differences between delivery delays. In cases where delivery ratios (Figure 4.7) are (close to) zero, the delivery delay results are less reliable.



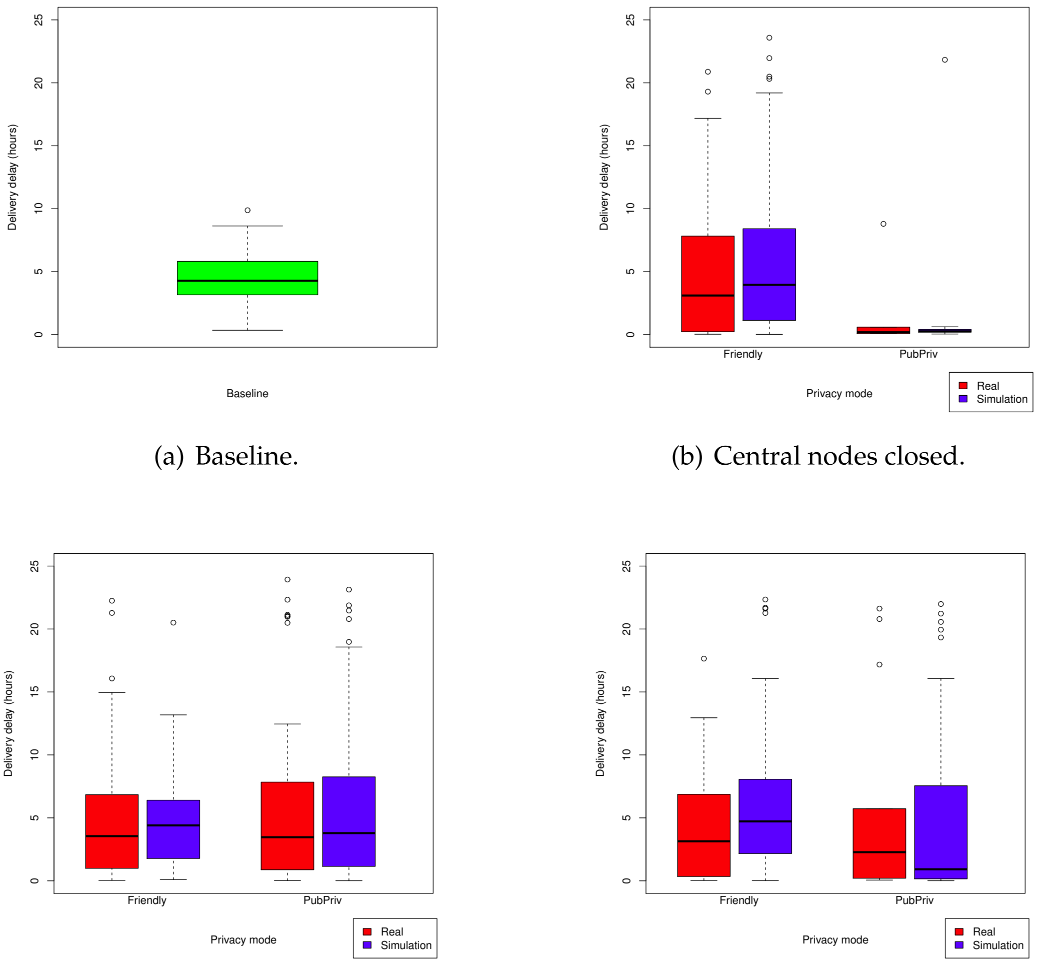
(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.9: *LocShare* dataset: Delivery ratio baseline, and delivery ratios for the real and simulation groups under two privacy models and three allocation methods. Delivery ratio performance trends for the sparse *LocShare* dataset are similar to the also-sparse *Reality Mining* dataset (Figure 4.7). In particular, for the PubPriv mode under the central nodes closed allocation scheme, performance falls to zero.



(a) Baseline.

(b) Central nodes closed.

(c) Central nodes open.

(d) Random.

Figure 4.10: *LocShare* dataset: Delivery delay baseline, and delivery delays for the real and simulation groups under two privacy models and three allocation methods. Similarly to the *Reality Mining* dataset (Figure 4.8), delivery delays show wider variation under the privacy modes than the baseline, but are generally similar under each privacy mode. The exception is the PubPriv mode with central nodes closed. Paradoxically, delivery delay is reduced; as for the synthetic model (Figure 4.3(b)) this is an artifact of the correspondingly low delivery ratio.

Algorithm 2 Decision: exchange messages during encounter

```
procedure Encounter($node_A$, $node_B$)
{The weighting of the $privacy\_behaviour$ random choice is specified by the privacy model. Random weighting according to each node's previously allocated category.}
$privacy\_behaviour[node_A] \leftarrow$ random choice of {nobody, friends, everyone} weighted by weightings for $node\_category[node_A]$
$privacy\_behaviour[node_B] \leftarrow$ random choice of {nobody, friends, everyone} weighted by weightings for $node\_category[node_B]$
$overall\_privacy\_behaviour \leftarrow$ more restrictive privacy choice of $node_A$ or $node_B$
if $mode ==$ Friendly then
  if ($overall\_privacy\_behaviour ==$ everyone) || ($overall\_privacy\_behaviour ==$ friends && $are\_friends(node_A, node_B)$) then
    messages are exchanged according to SRSNR (i.e., each message is forwarded if the encountered node is a friend of the original message sender)
  else
    no messages are exchanged
else if $mode ==$ PubPriv then
  if $overall\_privacy\_behaviour ==$ everyone then
    messages are exchanged according to SRSNR
  else
    no messages are exchanged
```

tion simulation group, and lower for the PubPriv mode than the Friendly mode, with differences greater between modes than between groups. In this densely-connected dataset, there is little performance impact in each case when changing node allocation method.

The *Reality Mining* dataset is sparser than the *SASSY* dataset, with lower absolute delivery ratios as shown in Figure 4.7. Compared to the baseline, delivery ratios are significantly lower for the Friendly privacy mode, and lower still for the PubPriv mode — falling to zero for the PubPriv mode when using central nodes closed or random node allocations. In each case, delivery ratios are similar for the real and simulation groups: there is overlap in the boxes. Figure 4.8 shows that there are no significant differences between delivery delays, however: any differences are lost in the noise. An exception is that delivery delays are less reliable for the cases where delivery ratios (Figure 4.7) are (close to) zero.

Similarly to the *Reality Mining* dataset, the *LocShare* dataset is also sparse. *LocShare* delivery ratio performance trends, shown in Figure 4.9, are similar to those for *Reality Mining* (Figure 4.7). In particular, we note that for the PubPriv mode under the central nodes closed allocation scheme, performance again falls to zero — implying that the network is useless. Delivery delay results for *LocShare* (Figure 4.10), similarly to *Reality Mining* (Figure 4.8), show wider variation under the privacy modes than the baseline, with any performance differences lost in the noise. The exception is the PubPriv mode with the central nodes closed allocation scheme; paradoxically, as a consequence of the low delivery ratio, delivery delay is reduced, as we have seen previously for the synthetic privacy model (Figure 4.3(b)).

4.5 Summary

We have presented a methodology to use empirically-measured privacy concerns from a simulated application in order to simulate the potential performance impact on future opportunistic networks. The empirical privacy models that we present have been developed to be dataset-independent, and we have applied them to three real-world traces.

The simulation results demonstrate that location privacy concerns may have a significant impact on network performance. This applies both for a synthetic privacy model and for an empirical privacy model. In the worst case, we find that the opportunistic network performance (as measured by delivery ratio) may fall to zero, implying that the network is useless.

Might it be possible to alleviate privacy concerns by using privacy-preserving routing protocols? If so then this may improve participation in the network, which is paramount for performance. The next chapter examines this question, with a focus on maintaining the privacy of the social graph; as discussed in Chapters 2.3.1

and 4.2, social graph privacy and location privacy are entwined — improvements to social graph privacy can therefore help to maintain (and thus allay concerns relating to) location privacy.

Chapter 5

Privacy-enhanced social network routing protocols

5.1 Introduction

We have seen in Chapter 4 that privacy concerns may have an impact on opportunistic network performance, as users reduce their participation in the network in order to preserve their privacy. In this chapter, we consider mitigating a passive attack on privacy — against privacy of social contacts — through the use of privacy-enhanced routing protocols. If the routing protocol maintains privacy, then users may be more inclined to participate in the network as their privacy concerns are alleviated.

In some simple social network routing schemes, such as SRSN Routing (SRSNR), the sender's friends list is transmitted in the clear along with each message. Intermediate forwarding nodes are able to read the sender's full friends list in plain text.

Encrypting the friends list end-to-end can ensure privacy, but we would then lose the advantages of social network routing: intermediate forwarding nodes would no longer be able to exploit the sender's social network information to inform their routing decisions.

If friends lists could be encrypted using pair-wise keys shared between each pair of nodes, then an eavesdropper could not overhear the sensitive data. This has two problems, however. Firstly, as discussed in Chapter 3.2, we do not assume the existence of a public key infrastructure (PKI) in opportunistic networks — due to the nature of such networks, we regard building a PKI as extremely difficult at best, and arguably impossible. Secondly, and more fundamentally, the sender may not wish to broadcast their social network information to all of their contacts — which would necessarily occur for this information to be used by these contacts as intermediaries for routing.

As discussed in Chapters 2 and 3, nodes may not have global knowledge about the network. Therefore, inspired by [11], in this chapter we attempt to target the privacy threats introduced by social network routing by modifying and obfuscating each sender's friends list, on a per-message basis, at message generation time. This may be performed locally at each node, without requiring any global knowledge of the network. By modifying the friends lists, we aim to introduce plausible deniability; each list transmitted is no longer a true copy of the friends list. By obfuscating the friends lists, we aim to make it more difficult for a person with a copy of a particular friends list to read out its contents.

The contributions of this chapter are:

- We introduce two novel routing protocols to enhance privacy in social network routing without key management.

- We evaluate the performance of these protocols against real-world datasets, and demonstrate that it is possible to obfuscate the social networking information without a significant decrease in routing performance.

- We discuss the privacy gains in using these protocols, with reference to the classes of attack that are mitigated.

5.2 Privacy-enhanced routing protocols

We introduce two novel privacy-enhanced routing protocols. In these protocols, to preserve privacy the sender modifies and obfuscates the copy of their friends list included with each message, on a per-message basis, at message generation time.

5.2.1 Statisticulated Social Network Routing

Named for a portmanteau of *statistical manipulation*,[1] our first scheme is *Statisticulated Social Network Routing* (SSNR).

For each message transmitted, the sender perturbs the message's copy of their friends list — adding or removing nodes. While the friends list sent along with the message will be based to some extent on the sender's true friends list, and so still useful for social network routing, the friends list has been modified by the addition or removal of nodes. Any node seeing the friends list sent along with the message now cannot say with certainty whether a particular node is truly one of the sender's friends, or truly not one of the sender's friends.

The sender may in practice choose the level of manipulation of the friends list on a per-message basis. In our evaluation, however, we examine routing performance for a particular choice of modification degree of the sender's friends list. For instance, we may choose a +50% modification of the friends list. This notation signifies that the sender adds 50% more nodes to their friends list before message transmission. We thus determine average performance for a particular degree of friends list modification. For simplicity, we do not evaluate routing performance while simultaneously adding and removing nodes; only for either adding or removing nodes.

It would still be possible for a malicious person to average over the friends lists included with many messages of one particular sender. But we have created much

[1] Huff coins the term *statisticulation* in his book *How to lie with statistics* [73], where he writes:

> "Misinforming people by the use of statistical material might be called statistical manipulation; in a word (though not a very good one), statisticulation."

more work for this malicious person: many generated messages must be intercepted, rather than just one single message to reveal all. We quantify and discuss the nature of the improvement in security in detail in Chapter 5.4.

5.2.2 Obfuscated Social Network Routing

In our second scheme, *Obfuscated Social Network Routing* (OSNR), instead of transmitting the sender's friends list as a list of nodes, we embed the friends list within a Bloom filter.

A Bloom filter [23] is a probabilistic data structure which allows probabilistic querying for set membership. False negatives are not possible, but false positives are — with increasing probability as the Bloom filter becomes more full. After inserting each node in the sender's friends list into a Bloom filter, we may regard the Bloom filter itself as a non-trivially-reversible hash of the friends list.

We are not the first to leverage Bloom filters for privacy: Schnell *et al.* [123] describe how similar records in a database can be linked using a Bloom filter, while maintaining privacy of each record. To our knowledge, we are the first however to use Bloom filters in the context of opportunistic network routing privacy. More recently, after we had published in [110, 111], Bianchi *et al.* have explicitly quantified the privacy enhancement associated with Bloom filter use in general [17].

To make a rainbow table attack [107] impractical, we create a per-message random salt, which is sent along with the message in the clear. The elements inserted into the Bloom filter are a concatenation of this random salt with a unique node identifier (any unique node identifier would suffice, e.g., a lower layer construct such as MAC address, Bluetooth address, IMEI, or some higher level identifier tied to the user rather than the device). In our evaluation, we choose to use Bluetooth addresses as the identifier.

Given the Bloom filter, the random salt (transmitted in the clear with the message) and an encountered node's identifier, it is easy to make a routing decision: query for set membership of the random salt concatenated with the candidate node identifier. A positive result — guaranteed if the candidate node is inside the sender's friends list, but also possible with low probability if not — means to forward the message, since the encountered node is most likely in the original sender's friends list. A negative result means that the candidate node is not in the sender's friends list, and so not to forward the message.

Since we are not using encryption (we assume no PKI), it still may be possible for an attacker to reverse engineer the Bloom filter by brute force — the attacker can iterate through all the node identifiers, concatenating each with the plain-text salt and testing for a Bloom filter match. This is orders of magnitude more work than a rainbow table lookup, however, and the brute force step must be repeated for every message. So using the Bloom filter (with salt) does not provide perfect security, but does make the attacker's job very much more difficult. We elaborate and quantify this attack further in Chapter 5.4.

It is possible to combine OSNR and SSNR: the friends list may be modified as in SSNR prior to hashing the social network information in a Bloom filter as in OSNR. In our evaluation, we refer to this combined scheme as SSNR-OSNR.

We note that Bloom filters are fixed-width — a convenient property for scalability. In pure SSNR, packet headers may grow arbitrarily large as the sender's friends list grows; this is potentially a problem for a sender with a very large social network (and compounded if the social network is grown further using SSNR). OSNR, and SSNR-OSNR, have no such scaling problem due to the fixed size of the Bloom filter.

5.3 Performance evaluation

We now present an evaluation of our two schemes to determine their impact on opportunistic network performance, using trace-driven simulation.

5.3.1 Datasets

We choose the same three datasets as used previously in Chapter 4.3.1 to evaluate our routing schemes: the *SASSY*, *Reality Mining*, and *LocShare* datasets. As previously mentioned, these datasets have different scales and structures.

We also include supplementary results from a fourth dataset, the *NUS* dataset [133]. This dataset details weekly recurring class schedules for 22,341 students at the National University of Singapore. Following the original dataset providers [132], we regard an encounter between two students as occurring when the students share a session, since the students are in physical proximity within the same classroom. We regard two students as having one another as social contacts, so having one another in each of their respective friends lists, if they share at least one session in the week. The full *NUS* dataset contains 12.3M encounters and 6.2M social network links. Due to memory constraints, we therefore sample a subset of students by two distinct processes, to derive two new, smaller datasets of 500 students each — *NUS-R* and *NUS-L*— which we then use for the performance evaluation:

NUS-R is obtained by randomly choosing 500 students from the full *NUS* dataset. Results from this dataset may reflect a real-world opportunistic network deployment, where only a proportion of students participate in the opportunistic network. A downside of random selection, however, is that due to the low percentage of selected nodes, the cutdown social graph would intuitively be expected to be relatively sparse, and so may exhibit different behaviour to the original more densely-connected graph.

NUS-L is, in contrast, obtained by "growing" a social network of 500 students, starting from an initial randomly-chosen student. Intuitively, the resultant social network is intended to exhibit more natural connectedness properties, similar to the full social network, while being size-reduced.

The full details of the two sampling processes are provided in Appendix A.1.4.

5.3.2 Simulation parameters

We performed trace-driven simulations using these datasets with the following parameters:

- 900 messages generated per simulation. Each message was unicast from a random sender to a random recipient from that sender's contacts list.

  - *SASSY*, *Reality Mining* & *LocShare*: We simulate 30 days (choosing a random 30 days in the case of *Reality Mining*), and generate 30 messages per day.
  - *NUS-R* & *NUS-L*: Exploiting the cyclic nature of the dataset, for each simulation we select one full week of six business days, each with 13 business hours,[2] beginning each simulation at a random time throughout the week. That is, we duplicate the first week to obtain an identical second week, then select out a random one-week period from these two weeks for each simulation run. We simulate 150 messages for each of the six business days; total 900 messages.

- Each message has a TTL of one day:
  - *SASSY*, *Reality Mining* & *LocShare*: 24 hours.
  - *NUS-R* & *NUS-L*: 13 business hours, reflecting one business day in the compressed business time representation of the original *NUS* dataset.
- 10 runs for each set of parameters
- *SSNR* obfuscation from -80% to +200% at 20% intervals.[3] When adding nodes to the sender's contact list, the nodes added are chosen from the pool of nodes present within the dataset (and within the given time slice from the dataset, in the case of *Reality Mining*).

For the *SASSY* dataset, which contains location information, we used a modified version of the ONE simulator [86], which included our augmented random waypoint model, to generate ns-2 traces. For speed, we used ns-2 rather than ONE for all of the simulations. For the *Reality Mining* and the two *NUS* datasets, which have no location information, we could not use ns-2, so instead used a Python program to parse the encounters and simulate message-passing.

[2] The original, raw dataset is "compressed" to "business hours", not real time. [132]

[3] For some messages, it may not be possible to continue adding or removing nodes to reach the target modification — if we reach the upper bound of all nodes in the dataset added, or the lower bound of only one node remaining in the sender's social network, we stop adding or removing nodes for this message. For simplicity, we do not simultaneously add and remove nodes from the social network.

5.3.3 Performance metrics

To evaluate our simulations, we use the widely-used metrics from [75], as noted in Chapter 2.2.1:

- *Delivery ratio:* the proportion of messages that were delivered, out of the total number of unique messages created.
- *Delivery delay:* the length of time taken for a message to reach its destination: the time between the time at which the message is first sent, and the time at which the message first arrives.
- *Delivery cost:* the total number of messages (including duplicates) transmitted, normalised by the total number of unique messages created.

When computing these metrics, we disregard messages which were directly transmitted from original sender to final receiver. In the *NUS* derived datasets, there is a high rate of such encounters — because we derive social network information from the encounters — and so leaving in these messages obscures the performance of (non-trivial) social-network routing, where messages reach their destination via at least one intermediary. For the other datasets, although incidence is not so high, we also disregard such messages so as to allow comparisons across datasets. The performance of social-network routing is therefore underestimated: higher delivery ratios and lower delivery delay would be achievable if we allowed such messages in our analysis.

5.3.4 OSNR implementation

Since the false positive rate of a Bloom filter depends on the length of the Bloom filter, and the number of elements in the Bloom filter,[4] but the number of elements in the Bloom filter greatly varies between datasets (since the sizes of the friends

[4] The Bloom filter false positive rate ε is approximately $(1 - e^{-kn/m})^k$, where k is the number of hash functions (in our case, $k = 4$ since we split the 128-bit MD5 hash into four 32-bit integers); n is the number of elements in the Bloom filter; and m is the length of the Bloom filter (in bits).

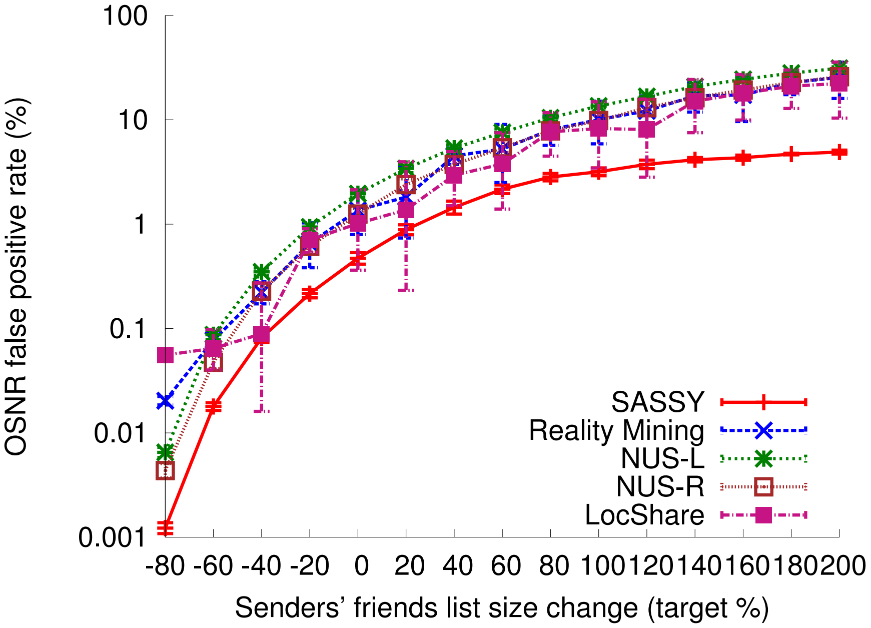


Figure 5.1: *OSNR* false positive rate for each dataset (data in Appendix C). The Bloom filter lengths were selected such that for pure *OSNR* (no *SSNR* modification), the false positive rate would be approximately 1%.

lists vary depending on the scale of the dataset), we choose the Bloom filter length on a per-dataset basis. We aim for a false positive rate of approximately 1% for the unmodified social-network routing case (0% *SSNR*) in each dataset.

Figure 5.1 (data in Appendix C) shows the actual *OSNR* false positive rate for each dataset, based on the average sizes of the routing friends lists and the lengths of the Bloom filters in each dataset (24 bits for *LocShare*, 32 bits for *Reality Mining*, 128 bits for *SASSY* and *NUS-R*, and 1024 bits for *NUS-L*). The higher the average size of the unmodified friends lists, the greater the length of the Bloom filter required in order to target an initial 1% false positive rate. To insert each element (string representation of a node ID concatenated with a random salt, as described in 5.2.2) into the Bloom filter, the element's 128-bit MD5 hash[5] was divided into four 32-bit por-

[5] MD5 is not collision-resistant, but we are only using the uniformity and one-way properties — not the collision-resistance property — of MD5. A maliciously-generated collision does not affect the security of our system, because the ability to generate a collision would merely result in another false positive in routing. Such false positives already occur with Bloom filters, and can more easily be triggered by manually setting more bits of the Bloom filter to 1 than by maliciously crafting an MD5 collision.

tions, each interpreted as a 32-bit integer. Taking each of these four integers mod the Bloom filter length L resulted in four values in range $0..(L-1)$. These four values were interpreted as indices for bits in the Bloom filter; and the corresponding bits were, if not already 1, set to 1.

5.3.5 Results

Figures 5.2–5.6 show our trace-driven simulation results for our routing schemes for each dataset (*SASSY*, *Reality Mining*, *LocShare*, *NUS-L*, *NUS-R*).

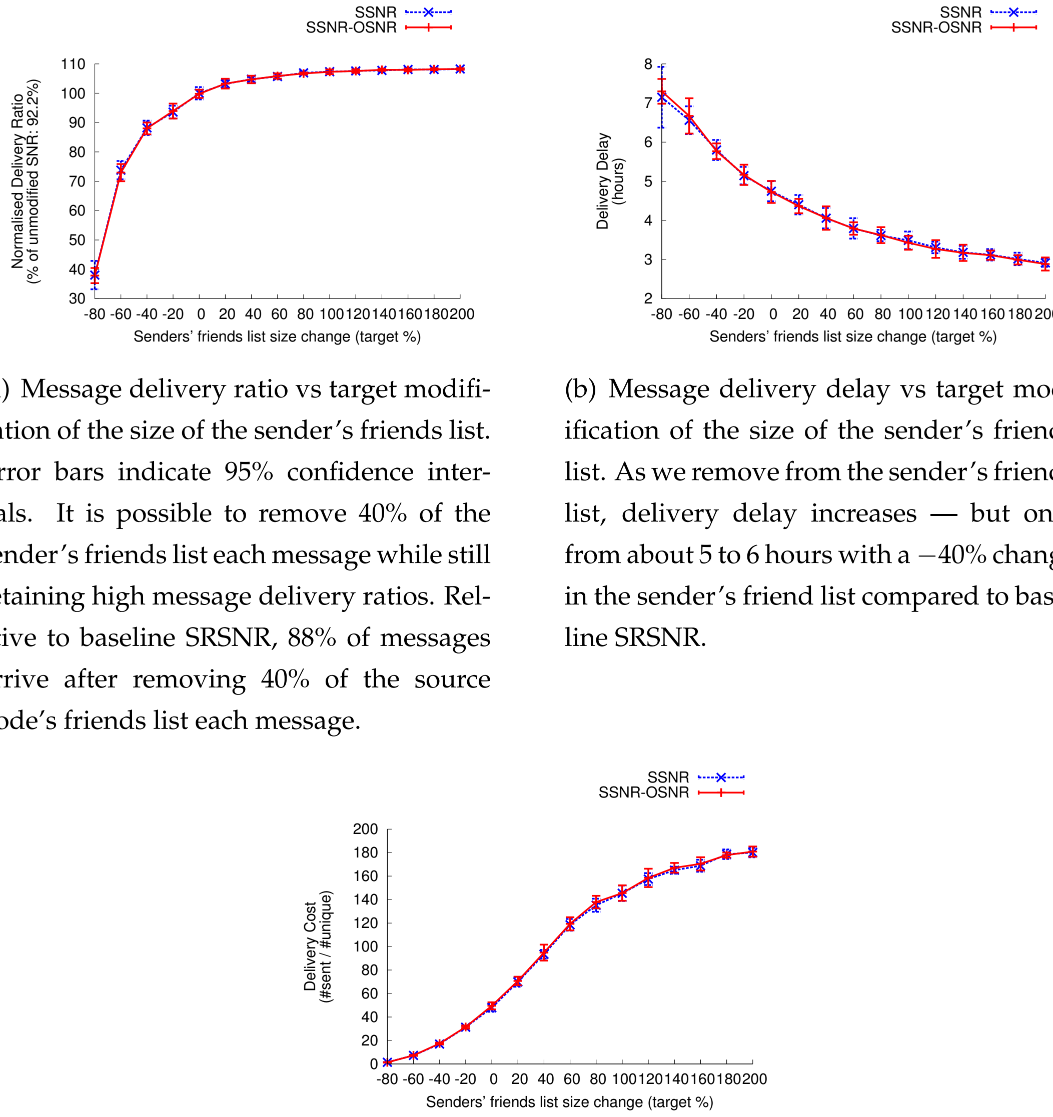
OSNR performance

We note that for every set of friends list size *reductions* for each of our three metrics, *the OSNR scheme did not significantly impact routing performance*.

Figure 5.1 offers an insight into why this may be: for pure *OSNR* (no *SSNR* modifications), the false positive rate was set — by choosing the length of the Bloom filter — to approximately 1%, as we discussed previously. This 1% false positive rate did not significantly affect routing performance, by any of our metrics. When removing nodes from the senders' friends lists, the false positive rate further decreases — and the decreased false positive rates also do not significantly affect performance by our metrics.

OSNR thus only ever had a noticeable impact on routing performance when *increasing* the size of the senders' friends lists. Even then this impact was often not significant (as for the *SASSY* dataset, shown in Figures 5.2(a)–5.2(b)).

On the few occasions when a significant difference between *SSNR* and *SSNR-OSNR* was visible — such as the upper end of friends list size increases for the *Reality Mining* dataset as shown in Figure 5.3(c) — we note from comparison to Figure 5.1 that the false positive rate for the Bloom filter had grown very high (30% for *Reality Mining* +200% *SSNR*).

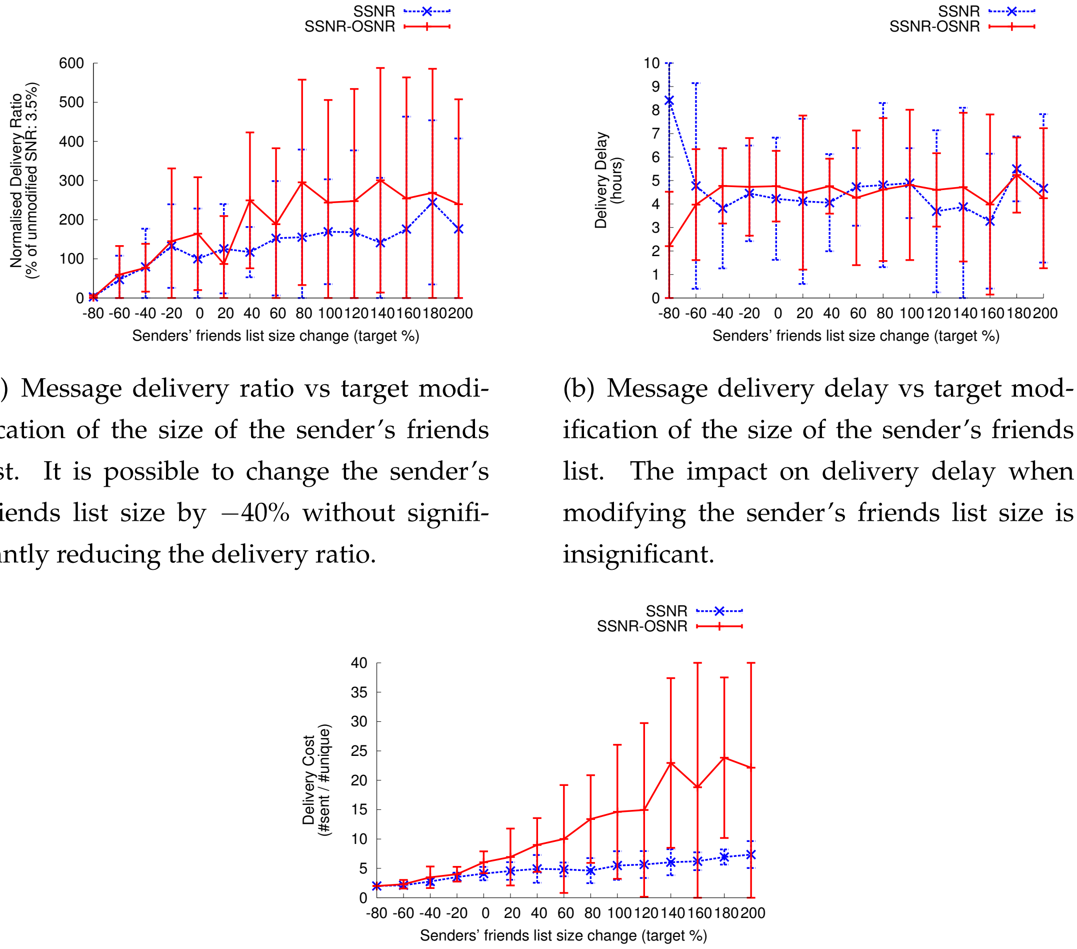


(a) Message delivery ratio vs target modification of the size of the sender's friends list. Error bars indicate 95% confidence intervals. It is possible to remove 40% of the sender's friends list each message while still retaining high message delivery ratios. Relative to baseline SRSNR, 88% of messages arrive after removing 40% of the source node's friends list each message.

(b) Message delivery delay vs target modification of the size of the sender's friends list. As we remove from the sender's friends list, delivery delay increases — but only from about 5 to 6 hours with a −40% change in the sender's friend list compared to baseline SRSNR.

(c) Message delivery cost vs target modification of the size of the sender's friends list. As we perturb the sender's friends list by adding links, the delivery cost increases.

Figure 5.2: *SASSY* dataset.

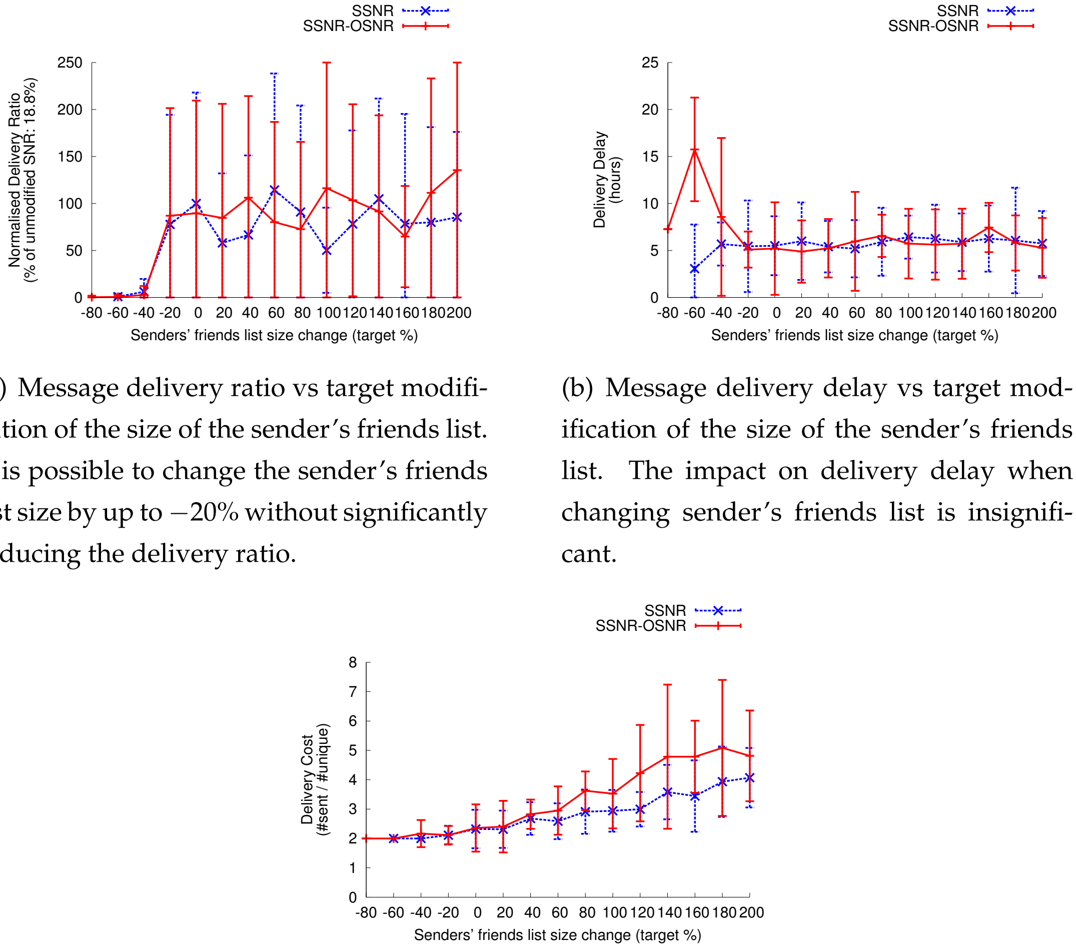


(a) Message delivery ratio vs target modification of the size of the sender's friends list. It is possible to change the sender's friends list size by -40% without significantly reducing the delivery ratio.

(b) Message delivery delay vs target modification of the size of the sender's friends list. The impact on delivery delay when modifying the sender's friends list size is insignificant.

(c) Message delivery cost vs target modification of the size of the sender's friends list. As we perturb the sender's friends list by adding fake friends, the delivery cost increases. When using *OSNR*, the false positives associated with using a Bloom filter also lead to an increase in delivery cost.

Figure 5.3: *Reality Mining* dataset.

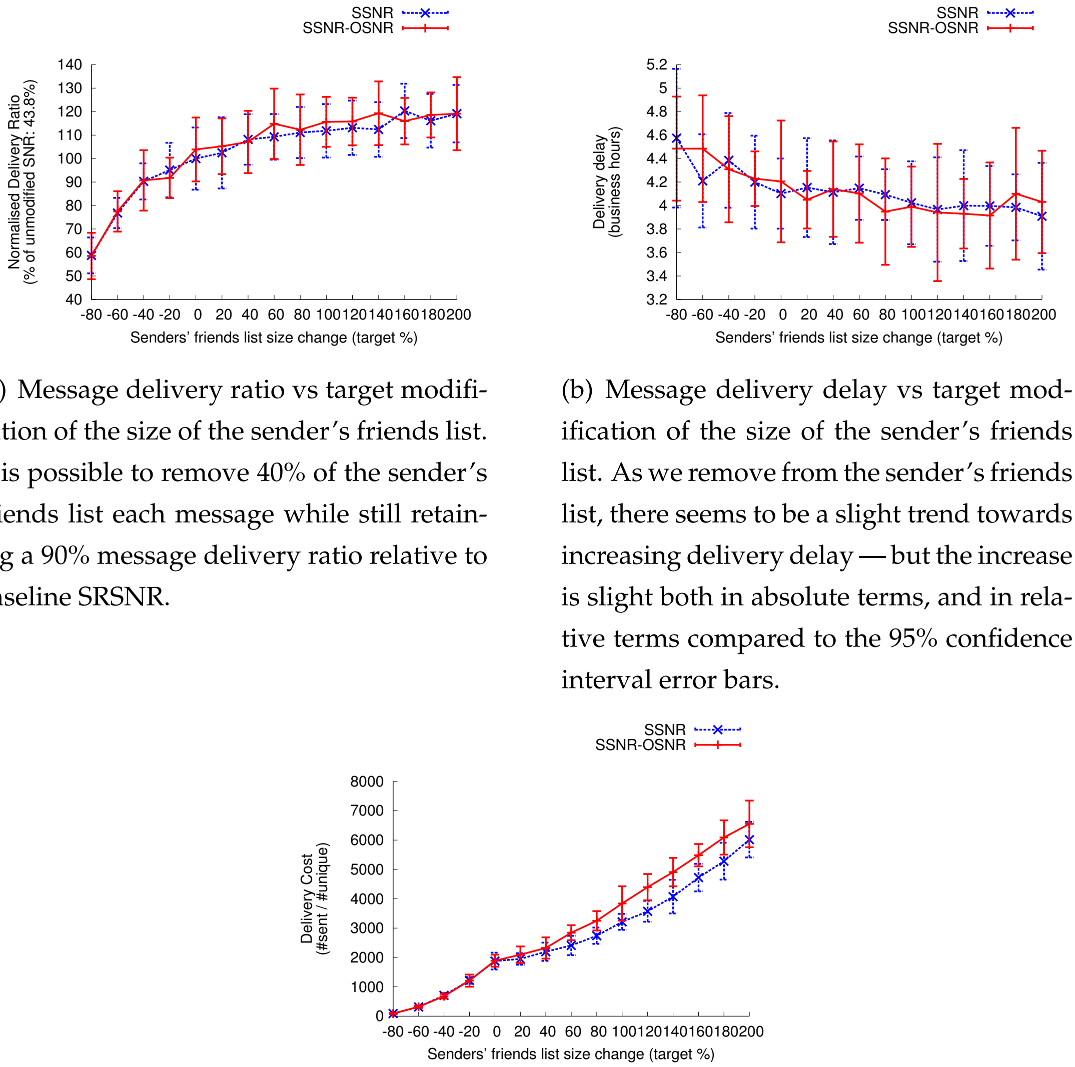


(a) Message delivery ratio vs target modification of the size of the sender's friends list. It is possible to change the sender's friends list size by up to -20% without significantly reducing the delivery ratio.

(b) Message delivery delay vs target modification of the size of the sender's friends list. The impact on delivery delay when changing sender's friends list is insignificant.

(c) Message delivery cost vs target modification of the size of the sender's friends list. As we perturb the sender's friends list by adding fake friends, the delivery cost slowly increases.

Figure 5.4: *LocShare* dataset.

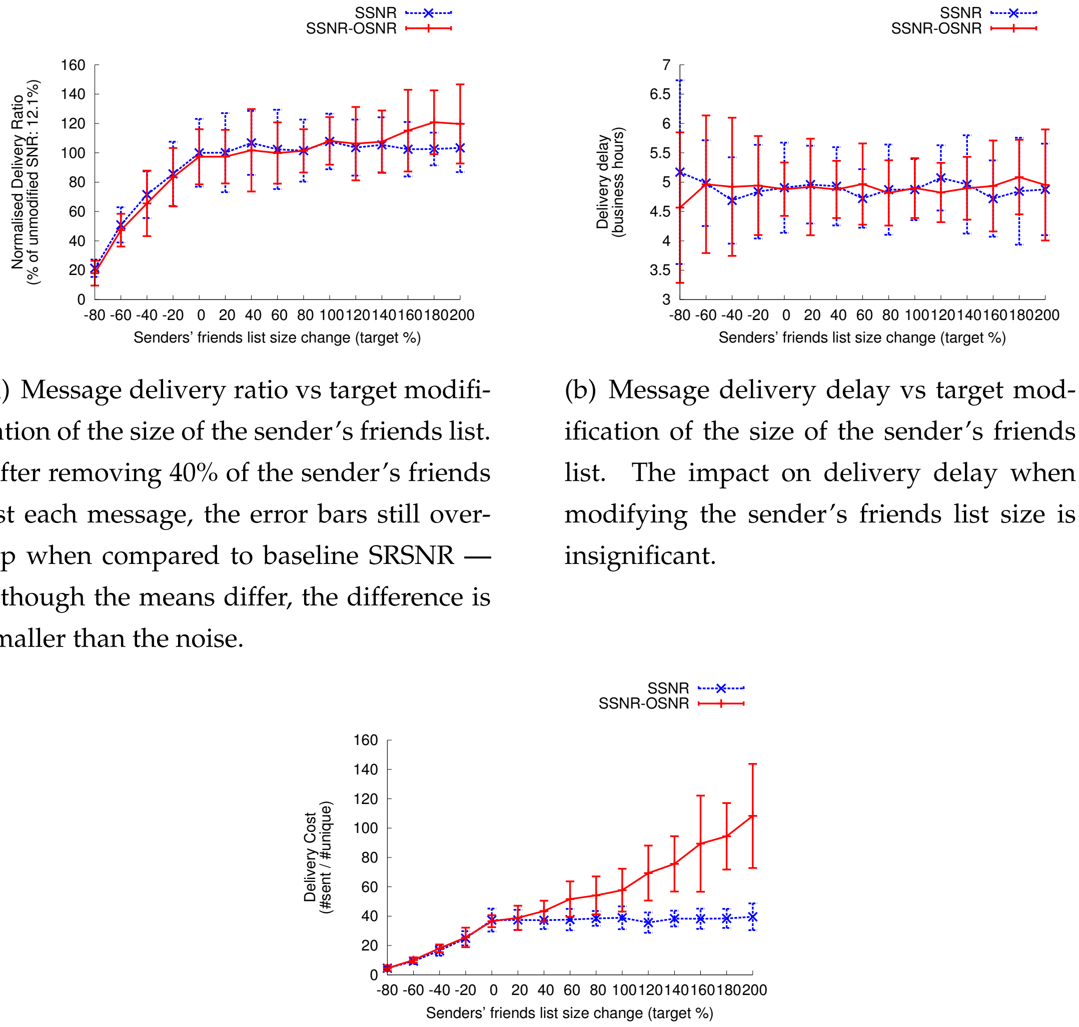


(a) Message delivery ratio vs target modification of the size of the sender's friends list. It is possible to remove 40% of the sender's friends list each message while still retaining a 90% message delivery ratio relative to baseline SRSNR.

(b) Message delivery delay vs target modification of the size of the sender's friends list. As we remove from the sender's friends list, there seems to be a slight trend towards increasing delivery delay — but the increase is slight both in absolute terms, and in relative terms compared to the 95% confidence interval error bars.

(c) Message delivery cost vs target modification of the size of the sender's friends list. As we perturb the sender's friends list by adding links, the delivery cost increases.

Figure 5.5: *NUS-L* dataset.



(a) Message delivery ratio vs target modification of the size of the sender's friends list. After removing 40% of the sender's friends list each message, the error bars still overlap when compared to baseline SRSNR — although the means differ, the difference is smaller than the noise.

(b) Message delivery delay vs target modification of the size of the sender's friends list. The impact on delivery delay when modifying the sender's friends list size is insignificant.

(c) Message delivery cost vs target modification of the size of the sender's friends list. As we perturb the sender's friends list by adding fake friends, the delivery cost does not significantly change when using pure *SSNR*, but does increase when using combined *SSNR-OSNR*. This may be because of the high false positive rate, c.f. Figure 5.1. When removing from the friends list, the delivery cost decreases for both schemes (pure *SSNR* and *SSNR-OSNR*).

Figure 5.6: *NUS-R* dataset.

SSNR performance

Delivery ratio Figure 5.2(a) shows that for the *SASSY* dataset, the delivery ratio is high for all tested social network size target modifications. It is possible to remove 40% of the nodes from the senders' friends lists, while still retaining a good delivery ratio: almost 90% of the ratio when not modifying the social network at all. Note that it is possible for the delivery ratio to *increase* when adding nodes to senders' friends lists (i.e., for a delivery ratio of over 100% that of using unmodified friends lists); by adding such nodes we are moving closer towards the best-case delivery ratio scenario of epidemic routing, where messages are forwarded during all encounters.

Figure 5.5(a) and Figure 5.6(a) show similar results for the two datasets derived from *NUS*. Although the absolute figures for delivery ratio are different due to the differing relative connectedness of *NUS-L* and *NUS-R*, the trend for the normalised delivery ratios are similar. Picking out the same -40% *SSNR* modification of the senders' friends lists, we note that, although the means differ (90% of unmodified social-network routing for *NUS-L*; 70% of unmodified social-network routing for *NUS-R*), the difference is smaller than the noise: the error bars overlap.

Although noisier, and with much lower absolute delivery ratios, than the other datasets, we see a similar result holds again for the delivery ratios in the *Reality Mining* dataset in Figure 5.3(a). It is possible to make large modifications to the sizes of the senders' friends lists without significantly affecting the delivery ratio, relative to the error margins.

For the *LocShare* dataset, however, Figure 5.4(a) shows that delivery ratios do fall sharply beyond -20% changes in friends list size. We believe that this is an artifact of low starting size of the friends lists (in absolute terms) of each node in this dataset, as shown in Appendix A.2: an already-sparse dataset is made sparser still, reaching a tipping point in delivery ratio performance.

Delivery delay Figure 5.2(b) shows that for the *SASSY* dataset, delivery delay increases somewhat when removing nodes from the sender's friends list. This increase is only from about 4.8 to 5.8 hours when using -40% *SSNR* compared to unmodified social-network routing. Indeed, if delivery delay is a concern, we may also reduce delivery delay by adding nodes with *SSNR*.

For the other datasets, the difference is typically not significant: Figure 5.3(b) (*Reality Mining*); Figure 5.4(b) (*LocShare*); Figure 5.5(b) (*NUS-L*); Figure 5.6(b) (*NUS-R*) all show little (if any) correlation between delivery delay and modification of the senders' friends lists. If such a difference exists, it is smaller than the noise — the error bars overlap within each dataset for each set of *SSNR* parameters.

We note an artifact in the results: where delivery ratios are very close to zero (i.e., Figure 5.3(a) shows *Reality Mining* at -80% *SSNR* and Figure 5.4(a) shows *LocShare* at -40% *SSNR* and beyond), the corresponding delivery delay values take more extreme values.

Delivery cost Figure 5.2(c) shows that for the *SASSY* dataset, delivery cost is quite significantly affected by modifying the sender's target friends list size: the fewer nodes in the modified sender's friends list, the lower the cost of sending a message. Compared to baseline SRSNR, with 50 data messages per unique message, a -40% change in sender friends list results in only 20 data messages: fewer than half as many data messages. This corresponds to the result from Figure 5.2(a), where we still retain a high delivery ratio. By applying *SSNR*, we have actually improved performance by this metric, by reducing the delivery cost, but while simultaneously retaining a good delivery ratio — and increasing the sender's privacy by not revealing some of their true friends.

Figure 5.5(c) and Figure 5.6(c) show similar results (in relative, not absolute, terms) for delivery cost for the *NUS-L* and *NUS-R* datasets, when removing from the senders' friends lists. A -40% change in the size of these lists results in a more than halving of data messages for both datasets. The relative differences are large, though: for *NUS-L* the change in cost is from 1900 to 700, while for *NUS-R* the change in cost is from 37 to 17. The absolute difference is again presumably due to the differing degrees of connectivity of the datasets, as illustrated earlier in this section. The differing connectivity presumably also accounts for the differing performance of pure *SSNR* when increasing the senders' friends lists for these datasets: for the highly-connected *NUS-L* dataset, encounters with the "fake" new friends are likely, increasing delivery cost, while for the less-connected *NUS-R* dataset such encounters do not occur so much, keeping delivery cost about constant. We note that combined *SSNR-OSNR* does show an increase in delivery cost for *NUS-R*, because extra encounters do occur (and with high false positive rates for the Bloom filter result in message forwarding) — but the number of such extra encounters is high relative to the relatively-small size of the senders' friends lists (as shown in Figure A.2), and hence the absolute number of extra forwarding opportunities in pure *SSNR* is low, keeping the delivery cost unchanged.

Figure 5.4(c) shows little change in delivery cost for the *LocShare* dataset when removing from the senders' friends lists, since the baseline cost is close to the minimum cost that can be measured by these simulations (i.e., two copies of each message: one from sender to intermediate node, and one from intermediate node to destination). There is, however, a slow increase in cost on adding to the senders' friends lists.

Figure 5.3(c) shows that delivery cost for the *Reality Mining* dataset stays fairly constant (since it is so low in absolute terms) in applying *SSNR* which reduces the friends lists sizes: the delivery cost falls from five messages to three messages on applying -40% SSNR. A similar effect is seen on applying *SSNR* which increases the friends lists sizes as for the *NUS-R* dataset. The senders' friends lists are, in absolute terms, small, as shown in Figure A.2. So increasing the relative size of the friends lists does not dramatically change the delivery cost with pure *SSNR*, since few encounters occur with the added fake friends. When adding these fake friends on applying combined *SSNR-OSNR*, however, the false positive rate ends up high (as shown in Figure 5.1) — about 30% at the upper end of the scale. This means that extra messages are forwarded, as triggered by this high false positive rate. Hence the delivery cost increases.

Performance summary Finally, we observe that it is possible to significantly modify the size of the sender's friends list (for example, by -40% for all datasets except *LocShare*), thus increasing the privacy of the sender, and yet to still retain good routing performance. Indeed, removing nodes may significantly reduce delivery cost — a beneficial side effect while enhancing privacy. If delivery delay or ratio is more of a concern, conversely, *SSNR* allows adding nodes to improve performance by these metrics, again while enhancing privacy, though at the expense of increased delivery cost in this case.

We quantify the improvement in security in the next section. We use the -40% friends list size change in subsequent analysis, since this was the largest size change that allowed performance to be maintained in most of the datasets.

5.4 Security discussion

The simulation experiments in Chapter 5.3 demonstrate that our schemes are practical with respect to performance: we are able to obfuscate the friends lists used for routing without a large impact on opportunistic network performance. We now consider the practicality of our schemes with respect to security — we discuss the privacy gains in using our schemes, with reference to classes of attack which are mitigated.

5.4.1 Security of *OSNR*

We first consider the *OSNR* scheme — where we hash the friends list of the sender to a Bloom filter.

In a naïve SRSNR implementation, an attacker may read the sender's friends list in plain text from one single eavesdropped message, as might an intermediate node who has legitimately received a message for forwarding. By hashing the sender's friends list to a Bloom filter, we raise the bar for a curious, casual observer — such as one of the sender's friends who legitimately receives a message as part of social-network routing. Our scheme keeps honest people honest. But we also increase the effort required by a malicious attacker. By how much?

OSNR with single intercepted message

In this attack an attacker attempts to reverse the Bloom filter, i.e., deduce the sender's original friends list from the Bloom filter. The attacker does so by iterating through the universe of elements that may be contained within the Bloom filter, and testing the Bloom filter for membership of each of these elements. For example, if a Bloom filter contains (salted) elements concatenated with a 32-bit node identifier (address), then to reverse the Bloom filter one should test each of the 2^{32} (similarly salted) addresses, for presence in the Bloom filter.

To give an impression of the expected effort required for an attacker with contemporary hardware to reverse the Bloom filters used in Chapter 5.3, we tested how long it might take to iterate through the complete universe of 32-bit addresses on a server with two Intel Xeon L5320 processors (2x quad core at 1.86GHz). We were able to test approximately 2^{14} addresses (concatenated with salt) for presence in a Bloom filter per CPU core per second. Iterating through the universe of addresses took 58 CPU-hours. With a larger address space (e.g., Bluetooth uses 48-bit addresses), the expected effort required would be greater still.

Since Bloom filters guarantee no false negatives, all of the addresses encoded inside the Bloom filter would be found by such iteration through all possible elements Bloom filters produce false positives, however, with a known rate ε — e.g., we targeted $\varepsilon = 1\%$ in choosing the Bloom filter length in our experiments. Therefore, the addresses truly encoded in the Bloom filter would be lost in the sea of false positives: with 4.3B addresses, we would expect 43M false positives. Thus an attacker would find it difficult to accurately deduce a node's friends list from eavesdropping a single message.

OSNR with multiple intercepted messages

We now consider an attacker who can intercept multiple messages. In our *OSNR* scheme, each subsequent intercepted message would allow the attacker to reduce the set of false positives (size f) to a new subset of size $\approx \varepsilon \cdot f$, each round.

Therefore, for Bluetooth addresses (2^{32} possible addresses) and a false positive rate $\varepsilon = 1\%$, the expected number of false positives, f, after intercepting n distinct messages is $f = 2^{32} \cdot 0.01^n$.

To recover the original friends list of the sender, the attacker must intercept sufficiently many messages, n, that $f < 1$. Rearranging the previous equation, this is

$n = log_{0.01}(2^{-32}) \simeq 4.8 \simeq 5.$

So under the assumptions above, the attacker must intercept approximately five distinct messages in order to recover the sender's original friends list.

The bulk of the computational burden on the attacker is reversing the first intercepted message's Bloom filter. After that, the attacker need only test the exponentially-decreasing number of elements from the previous rounds against newly-intercepted Bloom filters.

Implications of combined *SSNR-OSNR*

The combined *SSNR-OSNR* scheme is able to mitigate the eavesdropping attack, but how many distinct messages must an attacker intercept in order to recover the original friends list of a sender who is employing *SSNR*?

Chapter 5.3 shows that *SSNR* allows us to randomly remove 40% of the sender's true friends list per message without a major degradation in social-network routing performance. Using -40% *SSNR*, the probability of each member in a friends list appearing in a given message is $1 - 0.4 = 0.6$. As the attacker intercepts a number of messages n, the number of messages x in which a given member of a friends list appears (with appearance being random per message) is therefore binomially distributed, $x \sim B(n, 0.6)$.

Figure 5.7 shows the probability of the attacker identifying each friends list node as n increases, according to different threshold values of x, again using -40% *SSNR*. Using pure *SSNR*, the threshold is $x \geq 1$ — there are no false positives. To identify 95% of friends list nodes, four messages must be intercepted.

In practice, though, we combine *SSNR-OSNR*. The false positive rate is now defined by the Bloom filter. Using a $\varepsilon = 1\%$ as in our previous discussion, a suitable threshold might be $x \geq 3$ — the attacker may be confident that a friends list node is truly identified if the node appears in three or more intercepted distinct messages. Using this threshold, Figure 5.7 shows that the attacker must intercept eight messages in order to identify 95% of friends list nodes.

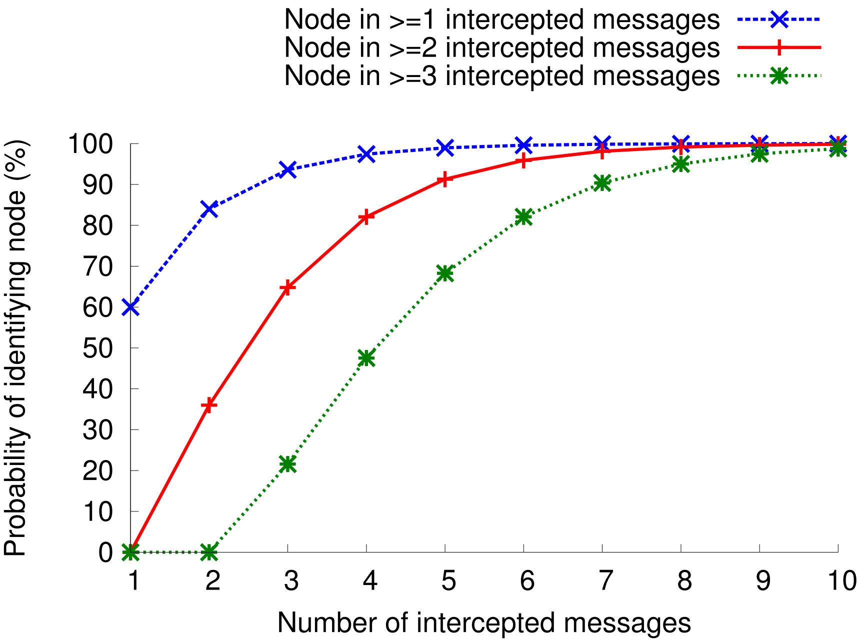


Figure 5.7: Probability of identifying each node within the sender's original social network after applying *SSNR* (-40%), as a function of the number of distinct messages intercepted. Using *SSNR-OSNR*, we consider the attacker as identifying a friends list node if that node appears in three or more distinct intercepted messages. To identify 95% of nodes, the attacker must intercept eight distinct messages.

Moreover, when combining *SSNR-OSNR*, the optimisation in Chapter 5.4.1 (discarding all addresses except those that were flagged up in previous rounds; initially mostly false positives but with the true addresses mixed in too) is also defeated since false negatives are now possible, further increasing the computational burden on the attacker.

Burden on the attacker

If an attacker must collect approximately eight messages in order to deduce the original sender's friends list, then how practical an attack is this?

Firstly, these messages need to be distinct. In *OSNR*, the Bloom filter is added by the original sender at the time of message generation, and is not altered en-route during routing. Therefore, distinct messages must be intercepted in order to obtain messages with different Bloom filters — it is not useful to capture the same message as it is routed through the network since the Bloom filter will be unchanged. Most opportunistic network implementations, however, are likely to employ bundle protocols which aggregate many application-layer data units into few network-layer data units for forwarding [57], thus hindering the eavesdropping of multiple messages.

To collect these messages, the attacker could shadow the sender, but if this were possible, then the attacker could directly observe the sender's interactions with other nodes and directly measure the sender's social network, rendering the attack redundant. An alternative strategy is to eavesdrop constantly in a well-known busy spot. Again, if this were possible, then an attacker could directly observe the social networks of many nodes.

Our schemes, therefore, are not infallible, but instead serve to raise the amount of effort required for an attack. Instead of being able to discover a sender's friends list by intercepting a single message and then reading off the data in plain-text, the attacker must now intercept multiple messages, and then devote multiple days of CPU time to the attack.

5.4.2 Linkability

While the bar for an attacker has been raised significantly for reversing a single sender's friends list, so too has the bar been raised much more for obtaining even a relatively small portion of the whole social network.

The structure of the social network itself is sensitive information, as we have discussed previously in Chapter 3.2.4. For example, as earlier noted, Narayanan and Shmatikov have shown that it is possible to link individuals who are members of different online social networks, based on no more information than anonymised node-edge graphs of both social networks [105]. Anonymity of social network participants is thus not sufficient for privacy, since the participants may be linked to another social network in which they also participate.

Narayanan's deanonymisation algorithm is described as being "robust to mild modifications of the topology such as those introduced by sanitization". This is because it deanonymises nodes by starting out at known seeds whose with positions known in both networks, and then crawling outwards from those seeds to find corresponding nodes in the two networks.

Thus, to be able to use this algorithm against an opportunistic network, an attacker would now have to be able to deduce accurate friends lists for a significant proportion of nodes close together in the social network. Crucially, the algorithm cannot "jump the gap" between disconnected subgraphs, so deducing the friends lists of some isolated nodes is not sufficient: the attack would only succeed if large-scale deduction of *all* the friends lists for nodes within a connected sub-graph of the social network were achieved by the attacker.

Such an attack would be difficult, but feasible. A single eavesdropped message would reveal the sender's complete local friends list information. Thus, by sniffing a sample of messages, the attacker may be able to gain enough information to reconstruct a fairly sizeable connected subgraph of the complete social network.

Our *SSNR-OSNR* scheme prevents this attack from being successful — or, at least, raises the bar very much higher for a potential attacker. The number of messages that must be sniffed is increased of the order of tenfold, since, as discussed in the previous section, approximately eight distinct messages from each sender must be intercepted to obtain the sender's local social network neighbours (their friends list) with some reasonable confidence. This must be repeated for each sender.

It therefore appears that our scheme may make these linkability attacks difficult, and, we believe, impractical.

5.5 Summary

In this chapter, we have presented two schemes to enhance privacy in social-network routing, which may be run locally at each node without requiring global knowledge about the network. The social network information is still used to inform routing decisions, but in a perturbed and obfuscated form.

We have seen that it is possible to perturb a sender's friends list by removing up to 40% of the nodes, while still maintaining a mean delivery ratio approximately 90% that of unaltered social-network routing. Complementarily, using Bloom filters we can mitigate eavesdropping of social network information with a minimal effect on network performance.

We have evaluated these two schemes using a selection of real-world opportunistic network datasets. Although these datasets vary widely in many properties (including scale, location and connectivity), our findings appear to hold for all, which gives us confidence that our schemes would be deployable in a real-world opportunistic network. We have also considered attacks against our schemes, and demonstrated the classes of attack which we may mitigate.

But another security threat for an opportunistic network is an active attack on availability, rather than a passive attack on privacy. In the next chapter, we explore the efficacy of such an attack, and how we might mitigate the threat through the use of a new lightweight, local protocol.

Chapter 6

Flooding attack mitigation using social network information

6.1 Introduction

In Chapter 5, we introduced two privacy-enhanced opportunistic network routing protocols. The protocols aim to protect potentially-private social network information from being easily passively eavesdropped, while maintaining social network routing performance.

But we have seen in Chapter 2.3.2 that opportunistic network users may also be exposed to active attacks. In particular, in Chapter 3.3 we noted a research gap in considering opportunistic network flooding attacks. We therefore focus on this particular type of flooding attack in this chapter.

Since it is difficult to determine reliably the sender of a message in an opportunistic network, a malicious user can untraceably flood the network with spoofed messages. As the available resources of participating devices (e.g., battery) are finite, and may be drained by receiving and retransmitting these messages, this flooding attack may therefore act as a denial-of-service attack against participating network nodes.

Our goal is to mitigate such a flooding attack, while maintaining the utility of the opportunistic network. The contributions of this chapter are:

- We formalise a flooding-based resource-consuming attack, and simulate the efficacy of the attack using real-world traces.
- We build a routing protocol that through using social network information is resistant to the attack.
- We demonstrate through trace-driven simulation that the attack-resistant protocol mitigates the attack, while at the same time maintaining the utility of the network.

6.2 Attack model and defence

Our goal is to investigate the impact of a flooding attack on an opportunistic network, and to mitigate this attack. In a flooding attack, the attacker floods the network with messages. Network nodes receive and relay copies of these messages throughout the network, consuming their finite resources (such as battery) in the process. The intent of the attacker is to overload these finite resources, causing nodes to fail, and consequently degrading overall network performance.

In order to formalise this attack, we consider an attacker with certain, limited capabilities, which we enumerate and formalise within the following attack model.

6.2.1 Attack model

As in previous chapters, we focus on SRSN Routing (SRSNR). Each node has a set of friends. The original sender of each message embeds a copy of their list of friends within the message, as part of its headers. This friends list then informs the routing of the message through the network: if a node appears in this list, then it will relay the message. For redundancy, the message is multiply copied, and thus may take more than one path to reach its destination.

We make the following assumptions, inspired by [30], about the capabilities of the attacker:

1. *Spoofing messages:* Messages are clear text, so the attacker can spoof any header of the message — or the entire message.

2. *Identity:* Nodes have some form of identifier within the network (e.g., MAC-layer address). In the absence of public key cryptography, this identifier can be spoofed on a per-message basis by an attacker.

Making these assumptions, the attacker may perform a simple flooding attack. When encountering another node, the attacker can generate a new message. This message, however, has spoofed headers, falsely indicating that it should be routed via the node — i.e., the node will believe that it is relaying the message on behalf of one of its friends.

Worse, the attacker can additionally spoof the "friends list" (i.e., the set of nodes which should relay the message) header, with a permissive set of nodes. This allows amplification of the attack: after the attacker injects the initial message into the network — by sending to the encountered node — the message will then be relayed, consuming further resources without additional cost to the attacker. This amplification is crucial to the attack: a relatively small number of messages generated by the attacker may be amplified many times throughout the network, thus consuming disproportionate network resources.

To further increase the attack, the attacker may spoof multiple MAC-layer addresses, in a manner similar to the Sybil attack [51]. This allow the attacker to send a larger number of messages to each encountered node: the node cannot blacklist a single MAC-layer identifier which generates numerous messages in a single encounter, because the messages appear to have been sent from numerous other encountered nodes.

Finally, the attacker may set an undeliverable destination address for the message. This ensures that the message will propagate as much as possible through the network (i.e., consuming greater resources), since it will never be delivered.

6.2.2 Defence

Due to the spoofing of headers, the above attack is difficult to detect at any node, using only its local knowledge. There is no way to determine a message's true origin. Therefore, even if a particular message should somehow be identified as an attack message, this lack of accountability and traceback means that only this one message would be locally dropped; the attacker may continue flooding other messages, under a new identifier.

We therefore introduce a new security requirement and assumption. The intention is to enable a lightweight scheme, where nodes authenticate that messages which they are willing to receive and relay are truly generated by one of their friends. We require a public/private key pair for each node. Each message is signed by its original sender, allowing any node knowing the sender's public key to verify the message origin.

One limitation of this scheme is that we require key distribution. As discussed in Chapter 3.2, a fully-fledged PKI may be unrealistic for a fully decentralised network. We note, however, that nodes have friends with whom they communicate, and as a prerequisite for SRSNR locally know who their friends are.

We therefore introduce a new assumption that *friends know one another's public keys*. These public keys may be shared between friends out-of-band of the opportunistic network without requiring a full PKI: possibly in a physical meeting, by earlier communication via traditional networking infrastructure, or even via snail-mail. Similar approaches to key exchange are used in some existing systems, for example Threema[1], a non-opportunistic mobile messaging application; security-conscious users may exchange keys in-person (scanning machine-readable QR codes), to enable later secure communication over untrusted networks.

Since messages in the network are only relayed by the original sender's friends, each relay node can thus verify that the message sender is truly their trusted friend by checking the signature (Algorithm 3): if the message is not signed by their friend, then it has been spoofed and is discarded. This mitigates the flooding attack.

[1] https://threema.ch/en/

Algorithm 3 Message check: only accept a message for relaying if the original message sender is a trusted friend.

```
if friends_with(message's original sender) and has_valid_original_sender_signature(message) then
    accept message for relaying
else
    discard message
```

It remains possible, however, for a node with genuine friendship links to other nodes to flood messages into the network; these messages will be authenticated and relayed by the attacker's friends. But this is a more expensive attack: the attacker must create genuine "friendship" relations with the nodes being attacked, and faking such a social relation is more expensive than spoofing a message. Additionally, even if a node can "trick" other nodes into becoming friends with it, the attack may still be mitigated. Each network node can now detect the attack, by looking locally at the messages which it has received for relaying. Each message can be linked back to its original sender. If a particular sender has generated excessive network traffic then this node can be blocked (i.e., blacklisted for relaying messages). This means that network nodes either (i) block the attacker locally, if the attacker has been successful in generating abnormally much traffic at that node, or (ii) do not see abnormal traffic from the attacker (perhaps due to the attack being throttled), in which case the attack is also unsuccessful. Either way, the attack is mitigated.

We note that other, more limited, wireless attacks may still be possible. For example, the attacker may attempt to overwhelm a single proximate node by transmitting invalid messages to it at a very high rate, as in a jamming attack [113]. The energy usage by the individual node to receive these messages — even if the messages are then immediately discarded as invalid — may drain its battery and take the node offline. But this is a weaker attack, i.e., without message amplification throughout the network. We consider targeted attacks on individual proximate nodes as out of scope for this chapter; our focus is mitigating flooding attacks with amplification.

Our proposed scheme relies on leveraging trusted social contacts. Using trusted social contacts to improve security in DTNs has been described by El Defrawy *et al.*, but in the context of preserving privacy rather than maintaining availability [55]. Whitelisting messages from immediate social contacts has been introduced in the context of email by Garriss *et al.* [63], and extended for more distant social contacts by Hameed *et al.* [68] — but both rely on a centralised architecture, and do not generalise to decentralised networks.

6.3 Evaluation

We now present an evaluation of the flooding attack on network performance, with and without the defence.

As in Chapters 4–5, we perform trace-driven simulation using a custom Python opportunistic-network simulator against the same three real-world datasets: SASSY, Reality Mining, and LocShare. Encounters and social network information are obtained for each dataset as discussed in previous chapters.

6.3.1 Simulation parameters

In line with the simulations detailed in previous chapters, we use the following simulation parameters:

- 100 runs per data point.
- One week of simulation time per run.[2]
- Average of one (non-attack) message per node per day.
- Message TTL of one day.
- Energy model for node batteries (following [18]):

[2] For *LocShare*, there are four one-week parts; we use each one-week segment with equal frequency. For *SASSY* and *Reality Mining*, we select one-week intervals where there are sufficient numbers of nodes present for non-trivial routing to be possible.

- Maximum energy: 1200 mAh; at the beginning of each simulation run, each node is assigned a random amount of energy between zero and this maximum.
- Energy loss per second: 1.9×10^{-3} mAh.
- Energy per message sent/received[3]: 0.4 mAh.
- Nodes participate in the network until they run out of energy. They then recharge offline for 8 hours, during which they do not participate in the network, and return with full energy.
- Infinite buffers; no transmission loss.[4]

As in previous chapters, messages which arrive in zero-time (i.e., from a direct link between the original sender and final recipient) are excluded from analysis; when sender and recipient are in proximity, there are presumably more efficient forms of communication than an opportunistic network. By excluding these transfers, we are able to focus on network performance in non-trivial opportunistic scenarios.

6.3.2 Flooding attack modes

Following [30], we pick one node from the trace in each run to act as the attacker; we do not add new attacker nodes because a model to generate synthetic node movement traces is beyond the scope of this thesis. The attacker attempts to flood the network with attack messages during encounters with other nodes. It does not participate in relaying background traffic.

We simulate the following modes:

- *Baseline*: As a neutral measurement of the behaviour of the network (i.e., without the attack being performed), no attack messages are generated. (Here,

[3] We assume that the same amount of energy is used per message for each of the modes introduced in Section 6.3.2, i.e., that the energy cost of signing a message or verifying a message signature is small in comparison to the fixed radio energy usage for exchanging the message.

[4] As detailed in Section 6.3.3, we focus on measuring message loss caused by overloaded nodes which run out of energy. We only introduce this one source of message loss to avoid confounding the results.

the "baseline" is with respect to the attack being performed, rather than with respect to a real-world traffic model.)

- *Vulnerable*: Simulation of the default behaviour of SRSNR, without any countermeasures to the flooding attack. At each encounter, the attacker generates 100 spoofed messages.[5] As discussed when introducing the attack, message headers are spoofed to ensure that the message is (i) undeliverable (i.e., has no real final destination node), and (ii) eligible to be relayed by any node.

- *Resistant*: Simulation of a passive defence scheme, as introduced in Section 6.2.2. The attacker must sign each attack message, so messages are only relayed via its genuine social contacts (friends). As for *Vulnerable*, the attacker sends 100 messages per encounter.

- *ResistantBlocks*: As for *Resistant*, but with an added active defence. Nodes locally maintain counts of messages they have received from each other node — with message origin verified since only signed messages from social contacts are accepted. Each node locally looks for any abnormal nodes, i.e., any node which has sent three standard deviations above the mean number of messages. If such a node is detected, then it is blocked at the detecting node; i.e., the node will discard further messages originating from this sender.

6.3.3 Metrics

To evaluate the efficacy of the attack, we use three metrics:

1. *Proportion of the time that (non-attack) nodes spend offline recharging*. For example, if each node spends eight hours in every 24 hours recharging, then it is offline recharging for 33% of the time.

2. *Delivery ratio*. The proportion of (non-attack) messages which arrive at their intended destination.

[5] Some traces have artifacts, where a single logical encounter is stored as numerous, consecutive physical encounters. For example, a Bluetooth scan may detect the same node during consecutive scans. To avoid skewing results due to these artifacts, we limit the attacker to sending to encountered nodes no more than once every ten minutes.

3. *Delivery delay*. The delay between the first transmission of a message, and its first arrival at its intended final recipient node.

If the attack is successful in overloading network nodes, i.e., causing them to run out of energy and fail, then we would expect the nodes to spend a greater proportion of time offline recharging. We therefore use as a metric the proportion of the time that the non-attack nodes spend offline recharging. The remaining two metrics — delivery ratio and delivery delay — are widely used as indicators of overall network performance [75].

6.4 Results

Figures 6.1–6.3 show our simulation results.

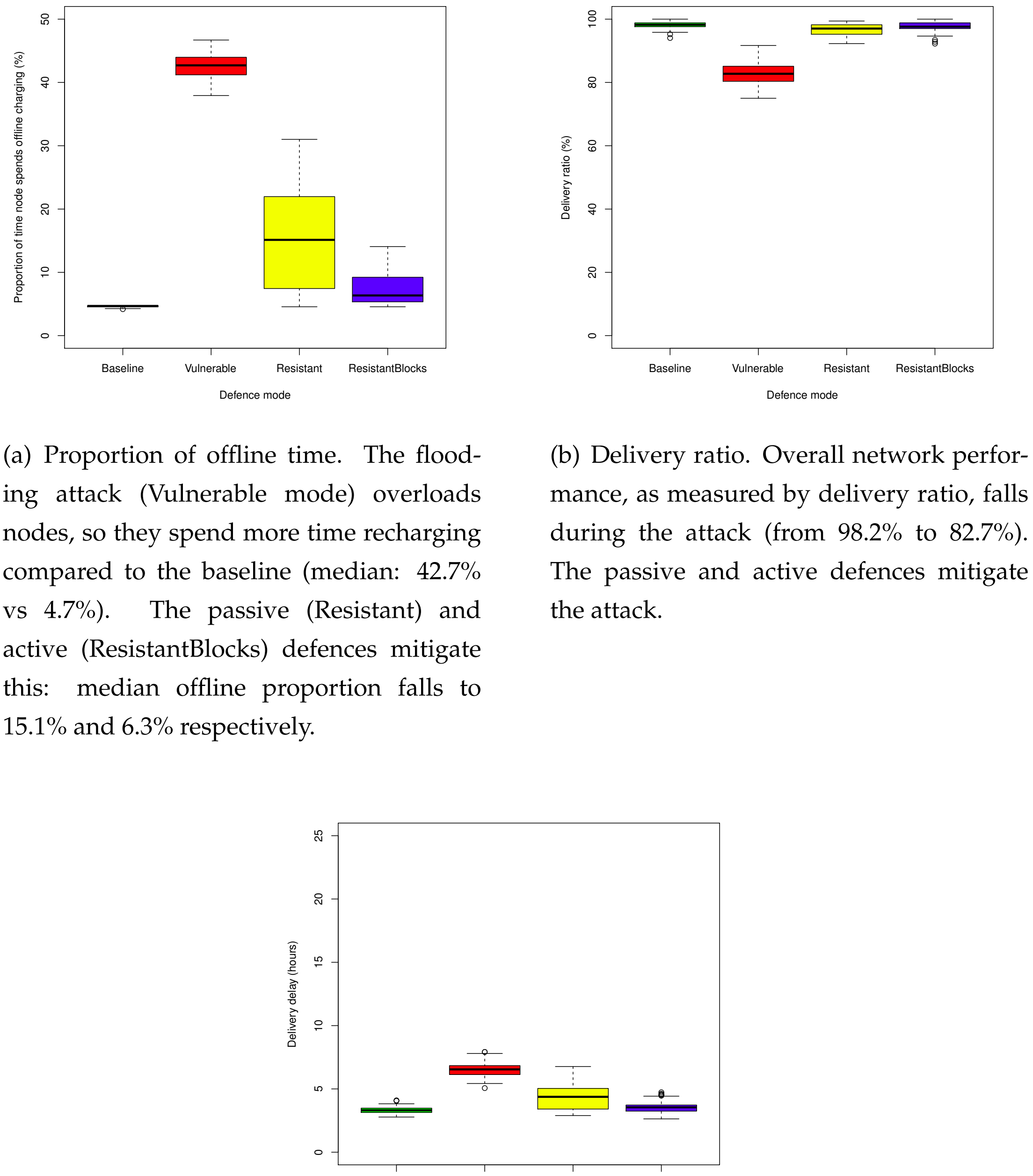
We consider two ways to measure the success of the attack: by examining the impact on individual nodes (with the metric of average proportion of time offline), and on the overall network performance (delivery ratio and delivery delay).

6.4.1 Impact of the attack

To determine the impact of the attack, we compare the metrics for each dataset in the *Vulnerable* mode, where the attack is performed, to the *Baseline* mode.

Figure 6.1(a) shows that there is a significant impact on nodes' proportion of time spent offline for the *SASSY* dataset. The attack has drained the nodes' energy, causing them to lose power. The median proportion of offline time is 42.7% for the *Vulnerable* mode, compared to the *Baseline* mode's 4.7%. For the sparsely-connected *Reality Mining* dataset, Figure 6.2(a) shows a more modest — but again significant — increase in node offline time, from 4.4% to 5.3%. The effect in the also-sparse *LocShare* dataset, Figure 6.3(a), is similar: 4.5% to 5.3%.

We have seen that individual nodes are affected. Is the network performance as a whole also impacted?

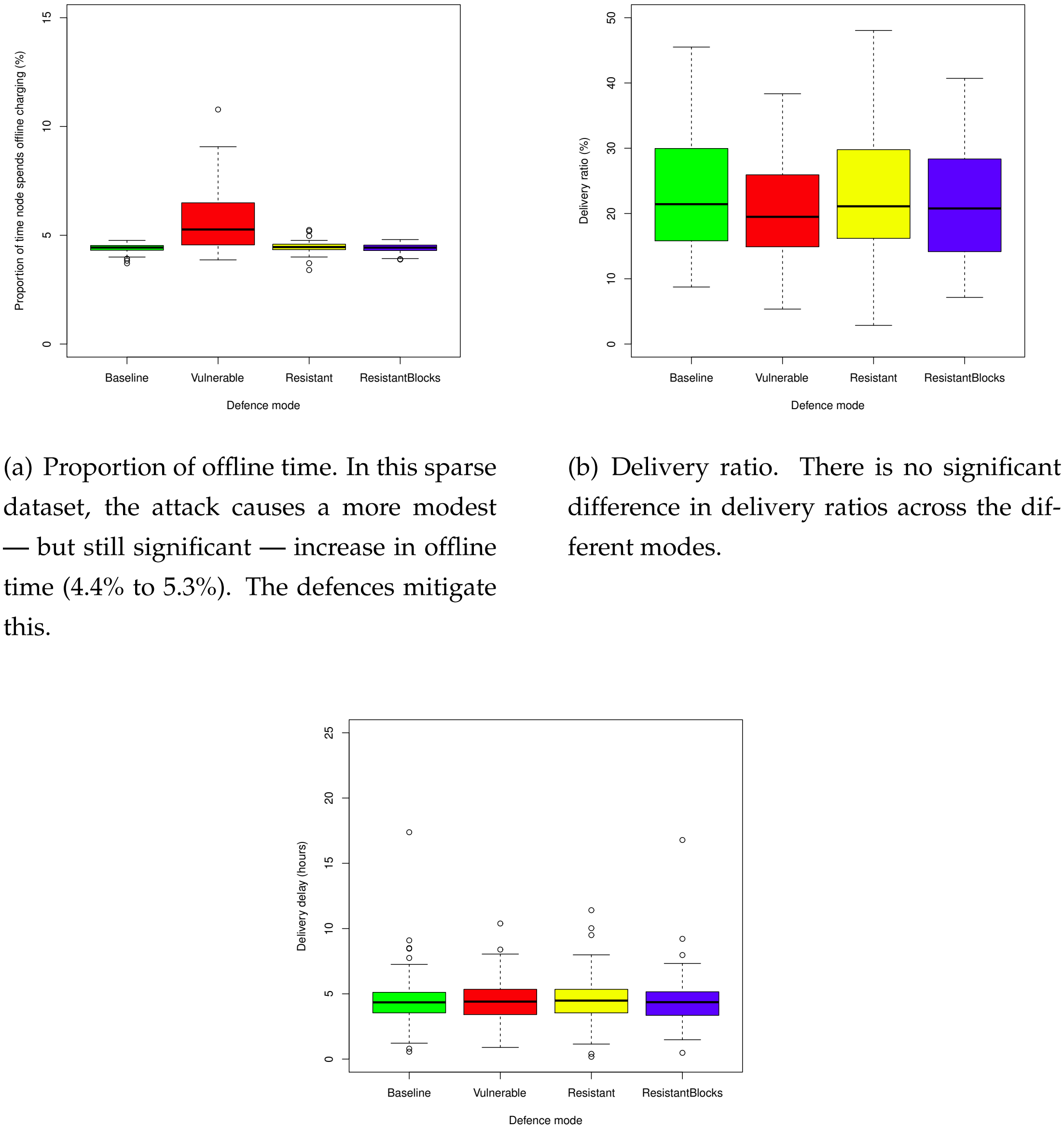


(a) Proportion of offline time. The flooding attack (Vulnerable mode) overloads nodes, so they spend more time recharging compared to the baseline (median: 42.7% vs 4.7%). The passive (Resistant) and active (ResistantBlocks) defences mitigate this: median offline proportion falls to 15.1% and 6.3% respectively.

(b) Delivery ratio. Overall network performance, as measured by delivery ratio, falls during the attack (from 98.2% to 82.7%). The passive and active defences mitigate the attack.

(c) Delivery delay. The attack worsens performance, increasing the delivery delay (3.3 hours to 6.5 hours). The passive and active defences mitigate this impact.

Figure 6.1: *SASSY* dataset.

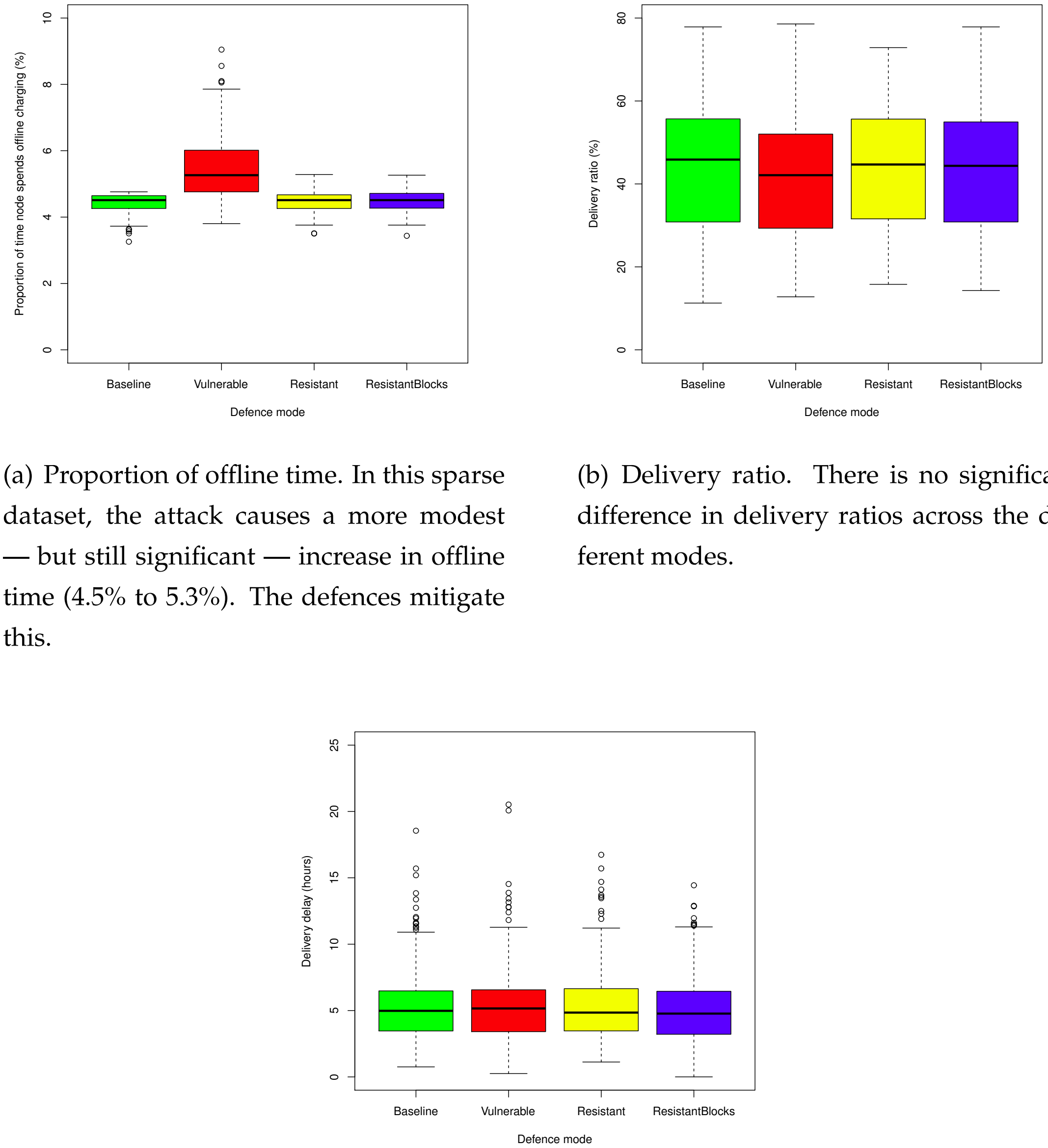


(a) Proportion of offline time. In this sparse dataset, the attack causes a more modest — but still significant — increase in offline time (4.4% to 5.3%). The defences mitigate this.

(b) Delivery ratio. There is no significant difference in delivery ratios across the different modes.

(c) Delivery delay. There is no significant difference in delivery delays across the modes.

Figure 6.2: *Reality Mining* dataset.



(a) Proportion of offline time. In this sparse dataset, the attack causes a more modest — but still significant — increase in offline time (4.5% to 5.3%). The defences mitigate this.

(b) Delivery ratio. There is no significant difference in delivery ratios across the different modes.

(c) Delivery delay. There is no significant difference in delivery delays across the modes.

Figure 6.3: *LocShare* dataset.

For the *SASSY* dataset, Figures 6.1(b)–6.1(c) show that there is a significant impact on the overall delivery ratio and delivery delay. The median delivery ratio falls from 98.2% to 82.7%, while delivery delay doubles from 3.3 hours to 6.5 hours. The *Reality Mining* (Figures 6.2(b)–6.2(c)) and *LocShare* (Figures 6.3(b)–6.3(c)) datasets, however, do not show a significant difference in network performance. We believe this is a consequence of the datasets' sparsity: the absolute delivery ratios are so low and variable that any impact is lost in the noise.

Summarising, the attack significantly increases offline times for individual network nodes. For the dense *SASSY* dataset, the overall network performance impact is also directly measurable. This demonstrates the efficacy of the attack.

We also note that impacting the energy of individual nodes (i.e., the mobile devices carried by network users) may discourage users' participation in the network. By theories such as Metcalfe's Law and Reed's Law [119], this may further reduce the value of the network for other nodes.

6.4.2 Efficacy of the defence

The *Resistant* mode implements the passive defence, and *ResistantBlocks* the active defence. By comparison to the *Vulnerable* mode, we can determine their efficacy.

From Figure 6.1(a), we can see that the defence effectively mitigates the effect of the attack on nodes' offline times for the *SASSY* dataset. Compared to a median of 42.7% time offline for the *Vulnerable* mode, this falls to 15.1% with the *Resistant* mode, and further to 6.3% with the *ResistantBlocks* mode — almost to the *Baseline* level. A similar trend holds for the *Reality Mining* dataset (Figure 6.2(a)) and for the *LocShare* dataset (Figure 6.3(a)). This is less pronounced, because the attack's impact was more moderate for these sparse datasets.

The network performance impact is also mitigated. For the *SASSY* dataset, Figures 6.1(b)–6.1(c) show an increased delivery ratio, and corresponding decreased delivery delay, using the *Resistant* mode (82.7% to 97.0%, and 6.5 hours to 4.4 hours). With the active defence, *ResistantBlocks*, the performance is further improved, to near-*Baseline* levels (97.6% delivery ratio, and 3.5 hours delivery delay). For the

Reality Mining and *LocShare* datasets, where the attack did not have a significant effect on network performance, the defence still does not worsen performance.

6.5 Summary

In this chapter, we have considered a flooding-based attack against opportunistic networks. We have demonstrated via simulation that the attack can impact opportunistic network performance, both at the global level with reduced delivery ratio and at the individual node level by draining nodes' energy.

We have further introduced schemes to detect and mitigate the attack, using only local social network information available at each node. We have demonstrated through simulation that the schemes appear effective.

The mitigation, however, depends upon a new assumption. Specifically, we assume the existence of some mechanism for out-of-band key distribution amongst socially-connected nodes. On the one hand, this may seem a reasonable assumption. If a node is "friends" with another node, then they may well have had sufficient opportunity to exchange keys prior to encountering each other in an opportunistic network scenario, for instance via meeting physically or through an infrastructure network. On the other hand, by requiring keys to communicate, we may be impeding potential uses of opportunistic communication. For instance, epidemic routing applications such as emergency broadcast or content distribution, where nodes send messages to any available node, are no longer possible. If epidemic routing is allowed, then a recipient node may no longer be able to verify a sender's key, which means that malicious nodes could generate throwaway public-private key pairs for forged nodes and so conduct the flooding attack.

Chapter 7

Conclusion

Opportunistic networks rely on participant nodes cooperating to forward data for one another. The decentralised and cooperative nature of such networks can expose network users to privacy and security threats, which in turn may discourage their participation in the network. We have therefore examined the following thesis:

> Privacy and security threats within opportunistic networks can be mitigated through cooperative social behaviour, without reducing network performance.

To test the thesis, we have considered the following questions:

Q1: How can we determine the performance impact of changes in behaviour due to users' privacy concerns, when we do not have a deployed opportunistic network?

Q2: Can privacy and security concerns that arise through the use of opportunistic networks be mitigated through cooperative social behaviour, while maintaining network performance?

To address the first question, we have presented a methodology in Chapter 4 to use empirically-measured privacy concerns from a simulated application in order to simulate the potential performance impact on future opportunistic networks. We have shown that users' location privacy concerns may have a significant impact on network performance.

To address the second question, we have presented a privacy-aware routing protocol in Chapter 5, and a security-aware routing protocol in Chapter 6. Each protocol relies on cooperative social behaviour, and is demonstrated to maintain network performance.

7.1 Contributions

In Chapter 4, we examined the performance impact of users reducing their participation in opportunistic networks due to a perceived threat to location privacy. We demonstrated using both synthetic and empirically-derived privacy models that user behaviour can have a large impact on opportunistic network performance. In the worst case, the opportunistic network performance (as measured by delivery ratio) may fall to zero, implying that the network is useless. This implies that to simultaneously respect users' privacy preferences and maintain network performance, privacy-enhanced routing protocols are required.

In Chapter 5, we considered how to create privacy-enhanced routing protocols, to preserve a second type of privacy: social graph privacy. We introduced two privacy-enhanced schemes — SSNR and OSNR — which can be used together. The schemes are based on perturbing and obfuscating the social network information used for social network routing, and can be used in the absence of key management. We quantified the privacy gains of the schemes, and demonstrated that they can be used while the performance of the network is maintained.

In Chapter 6, we demonstrated that it is possible to use social network information to mitigate one particular threat to security, specifically a flooding attack aimed at exhausting the finite energy of participating network nodes. We introduced an assumption, that social contacts ("friends") know one another's signing keys, and showed how to use this to authenticate messages passed through the network. We demonstrated that the flooding attack can be mitigated by such a protocol, thus maintaining or improving the opportunistic network performance.

7.2 Discussion

In Chapter 4, in order to investigate the impact of location privacy preferences on opportunistic network performance, we measure location privacy preferences for heavy Facebook users and assume that similar privacy behaviours apply for opportunistic network users. While we have argued in support of this assumption, we can imagine factors that suggest future research should examine validation of the assumption. For example, three factors which make affect privacy preferences are: (i) Facebook users are identified by name, while opportunistic network users are potentially pseudonymous; (ii) participants were prompted to choose privacy preferences for precise locations (up to the granularity of GPS), rather than for coarser locations as may be revealed in opportunistic network use; and (iii) opportunistic network users may be willing to accept a trade-off of some privacy in return for increased utility (e.g., improved network performance), which we would not be able to detect using this methodology.

Additionally, it may be possible to extend further the analysis that we have performed on the location privacy preference data collected in Chapter 4.4.3. For example, we can imagine more sophisticated privacy models, which take account of correlations between observed privacy preferences and other factors such as place or time-of-day: some places at certain times of day may be considered more sensitive information than others. Additional privacy modes could also be simulated for opportunistic networks, beyond those which we have described.

In Chapter 5, we detailed modified social network routing protocols intended to preserve privacy of opportunistic network users' social relations. As noted in Chapters 2.3.1 and 4.2, this can consequently help to preserve other types of privacy, such as location privacy, by hindering inference attacks. In Chapter 5.4, we quantified the privacy improvements associated with using the schemes, including discussing their limitations. In particular, we noted that the schemes keep honest people honest, and increase the effort required for a determined adversary, but are not infallible: given sufficient amounts of eavesdropped traffic, and computer time, the privacy protection can potentially be broken.

Such attacks are inherently possible when perturbing or obfuscating social network information on a per-message basis, as we proposed. It may be possible to hinder the attacks, however, by providing more consistently perturbed and obfuscated social network information. But this may introduce its own performance cost — for example, if a particular social contact is always removed, in every message, then no messages would be routed via that contact, which may produce a consistent decline in performance if the contact is central within the network. Alternatively, if we can assume some extent of key distribution — as we do in Chapter 6 — then it may be possible to increase privacy protection through encryption rather than obfuscation, since social network information could be encrypted during message exchange in order to thwart an eavesdropper. Even such a scheme, however, would not be infallible: a compromised node, or curious social contact, can still read the social network information used for routing — and more generally traffic analysis attacks may be possible.

In Chapter 6, we showed a mechanism to mitigate a flooding attack with a social network routing protocol. The scheme proposed necessitated an assumption, however: that a degree of key distribution is present, so that immediate social contacts ("friends") can verify the origin of one another's messages. We argued in support of this assumption, but testing the assumption would be challenging without a real opportunistic network deployment. Additionally, through requiring such an assumption, we may be impeding potential uses of opportunistic networks. For example, content could not be disseminated to non-friends while using such a scheme, since the core idea in the scheme is to accept messages from friends only.

Other considerations are that a more sophisticated attack may involve a trusted node that has been compromised; if the private signing key is compromised then it is possible to spoof message origin. The *ResistantBlocks* mode described in Chapter 6.3.2 could mitigate this to some extent, by detecting abnormal traffic appearing to originate from a "trusted" node — but at the expense of rejecting other messages from that node too, which may be authentic. Lower-level wireless attacks are also still possible, for example jamming or wireless flooding attacks. The scheme described does not mitigate such lower-layer attacks.

7.3 Future work

We conclude by noting potential new questions and avenues for future work, following on from the research presented in this thesis.

One natural extension would be to investigate combining the schemes presented in Chapters 5 (for preserving privacy) and 6 (for mitigating a flooding attack) into a unified opportunistic network routing protocol. Would it be possible to improve the privacy-preserving features of the protocol if we can assume, as in Chapter 6, that friends know one another's encryption keys? Or can we maintain the privacy defences while still being able to mitigate flooding attacks? We highlight these questions for future research.

A second natural extension would be to further investigate the relationship between location privacy (as was the focus in Chapter 4) and social graph privacy (as was the focus in Chapter 5). We discussed how these two types of privacy are entwined in Chapters 2.3.1 and 4.2; could we formalise this link further? In particular, would it be possible to quantify the extent to which location privacy concerns are allayed through improving social graph privacy — and in turn quantify the performance impact?

The focus in Chapters 4–6 has been on SRSN Routing (SRSNR). An additional avenue for research would be applying similar privacy and security techniques to other social network routing schemes, some additional examples of which we have discussed in Chapter 2. In particular, we note that SSNR relies on "all-or-nothing" binary friendship: two given users are either friends or they are not friends. A future area to explore would be whether we can find privacy-preserving or security-preserving schemes for social network routing protocols which involve graduated degrees of friendship, such as varying tie strengths, or which utilise indirect friends-of-friends.

While the simulations described in Chapters 4–6 rely on empirically-measured encounters and social network information, the traffic generation pattern is random, and not based on real data. Messages are unicast, between a randomly-selected sender and one of their randomly-chosen social contacts. Future work might investigate routing performance when using traffic patterns based on real-world data, or where some or all messages may be multicast.

We have focused in this thesis on a sample of specific privacy and security threats: location privacy in Chapter 4, social graph privacy in Chapter 5, and a particular flooding attack with message amplification in Chapter 6. This is not a comprehensive account of all privacy and security threats in opportunistic networks, leaving room for further research in this space. For example, one class of security problem that we have not considered in this thesis is selfishness. Nodes may wish to selfishly "freeload", by relying on other nodes to forward data on their behalf without participating in message forwarding themselves. Research is ongoing into adding incentives and reputations to opportunistic network forwarding (e.g., IRONMAN in [19]). A unified scheme — combining incentives, privacy-awareness and security-awareness across many types of threat — would be an ideal to strive towards in future research. A significant amount of future research is likely required in order to realise this.

Finally, we note that much opportunistic networking research in general — including that presented in this thesis — relies upon using relatively-few small-scale datasets. Would the results we have presented generalise to larger datasets, or particularly a real deployment? Could we obtain new insights by using larger datasets, or by studying a real deployment? We highlight such general questions too for future research.

Appendix A

Datasets

A.1 Dataset details

A.1.1 St Andrews mobile sensor network (*SASSY*)

The first dataset was collected at St Andrews from a deployed sensor network system, the St Andrews Sensing SYstem (SASSY), in a previous experiment by Bigwood *et al.* [20]. 25 participants were equipped with 802.15.4 Tmote Invent sensor motes and encounters were tracked for a period of 79 days, although for efficiency we chose to use only the first 30-day section of this trace for our simulations.

The original dataset was very sparse due to hardware limitations which meant that many encounters may have been missed. Inspired by [54] and [59], Bigwood augmented the collected traces using a working-day and augmented random-waypoint model. Nodes randomly select a waypoint from a set of points of interest and walk according to predetermined paths (such as roads) to reach these points. Nodes moved at 0.5–1.5ms^{-1}. At each waypoint the nodes could stop for 0–120s. Each node was additionally randomly assigned a home location, and the nodes would travel to this location to "sleep" for 8 hours in every 24. Each node had an additional 10% probability of either visiting the Computer Science departmental buildings (since our participants were mainly Computer Science students) or their "home" at any waypoint selection.

To obtain social network information for the *SASSY* dataset, we use the self-reported social network information provided by the 25 participants at the start of the experiment: their Facebook "friends". Many participants knew each other: the mean friends list size (i.e., number of Facebook friends also participating in the experiment) was 9.8, with a standard deviation of 5.0.

A.1.2 MIT Reality Mining (*Reality Mining*)

The well-known *Reality Mining* dataset [52] was collected at MIT [53]. This dataset comprises Bluetooth encounter traces from 97 mobile phone users over the course of an academic year. To obtain social network information for this dataset, we use the participants' address book information.

Although the *Reality Mining* dataset does not explicitly record participants' address books, we are able to infer address book information based on the included data. For each participant, the dataset includes a log of outgoing contacts (phone calls and SMS messages), along with both a pseudonymised contacted phone number and a flag to indicate whether this phone number is in the participant's address book. By matching the pseudonymised contacted phone number to that of another *Reality Mining* participant, we thus use the address book flag to determine whether the first participant has the second participant in their address book.

We construct the friends lists based on the address book information by defining that if at least one participant has the other in their address book, then the pair are said to be friends, i.e., each has the other in their friends list. Unlike the *SASSY* dataset, few participants knew each other: 52 participants had at least two other participants in their friends lists (and were thus candidate nodes for social-network routing in our simulations). Of these 52 participants, the mean friends list size is 3.7, with a standard deviation of 2.0.

Because of differing lengths of participation in the experiment,[1] we could not treat the dataset as one contiguous trace — it would not be meaningful to simulate message-passing between people no longer participating in the study. Therefore,

[1] Some participants carried mobile phones for the full nine months of data collection, while others participated for much lower amounts of time — as low as one month.

at the beginning of each simulation run, we draw out a random[2] 30-day segment of the trace.

A.1.3 *LocShare*

The *LocShare* dataset [14] was collected at St Andrews, as previously described in Chapter 4.4.1. While the lead researcher for collecting the dataset was Fehmi Ben Abdesslem, I contributed significantly (approximately equally) towards the experiment's implementation and collection of the raw data. I did not contribute towards collecting raw data for the other datasets used within this thesis.

We collected the locations of 80 participants during four one-week runs of 20 participants. By defining an encounter as occurring when two participants are within 10 metres — selected as this is the approximate average Bluetooth range — we obtain a trace of encounters between participants. As for the *SASSY* dataset, social network information was obtained from Facebook friendships.

While the primary purpose in collecting the dataset was to develop empirical privacy models (see Chapter 4.4.1), we also collected location data and social network information of participants. We therefore were able to construct a further dataset suitable for opportunistic network simulations, which we have made publicly available to other researchers [13].

A.1.4 NUS student contacts (*NUS*)

While the previous three datasets are the standard datasets used for evaluation throughout this thesis, we also include additional supplementary results in Chapter 5 only for an additional dataset, a student contact pattern dataset comprising the class schedules of 22,341 students at the National University of Singapore [133] (*NUS*).

[2] The only constraint placed on the selection of the random 30-day segment which we draw out is that at least three participants, each with at least two other participants in their friends list, must be carrying phones throughout the 30 day period — otherwise meaningful social-network routing could not occur (since there would be no message-passing intermediaries).

The dataset describes contacts between students in recurring weekly sessions. As in the original paper describing the dataset [132], we regard an encounter between two students as occurring when these students share a session — that is, when the students are in physical proximity in the same classroom. The assumption that data exchange is possible between two proximal mobile devices, even inside large classrooms, was experimentally validated in [132]. We regard two students as having one another as social contacts, so having one another in each of their respective friends lists, if they share at least one session in the week. Defining social links in this way, the mean friends list size is 561, with a standard deviation of 396.

Due to memory constraints, we do not perform simulations in Chapter 5 using the full *NUS* dataset: the full dataset contains 12.3M encounters and 6.2M social network links. Instead, we sample a subset of students from the full dataset in two different ways, to derive two new, smaller datasets. In each case, after extracting a subset of the students, we preserve all encounters and social network links between students within this subset.

1. We randomly select 500 students from the full 22,341 students in the *NUS* dataset. We call this derived dataset *NUS-R*. Results from this dataset may reflect a real-world opportunistic network deployment, where only a proportion of students participate in the opportunistic network.

2. A downside of randomly selecting students from the full *NUS* dataset is that doing so leads to a relatively sparse social graph. Therefore, we adopt the approach of Liu and Wu [96] to sample the *NUS* dataset in a second way, which avoids the extremes of sparsity (as occurs when randomly sampling students) or over-connectedness in the new, derived, size-reduced dataset. We select the first student randomly, and then, to select the k^{th} student, we randomly divide the previously-selected $k-1$ students into two equal-sized groups S_1 and S_2, and select the k^{th} student as that student with the highest $\sum_{s_1 \in S_1} sim(s, s_1) - \sum_{s_2 \in S_2} sim(s, s_2)$ where sim is the number of common class sessions in which two students are enrolled. Using Liu and Wu's approach, we sample 500 students from the full *NUS* dataset. We call this derived dataset *NUS-L*.

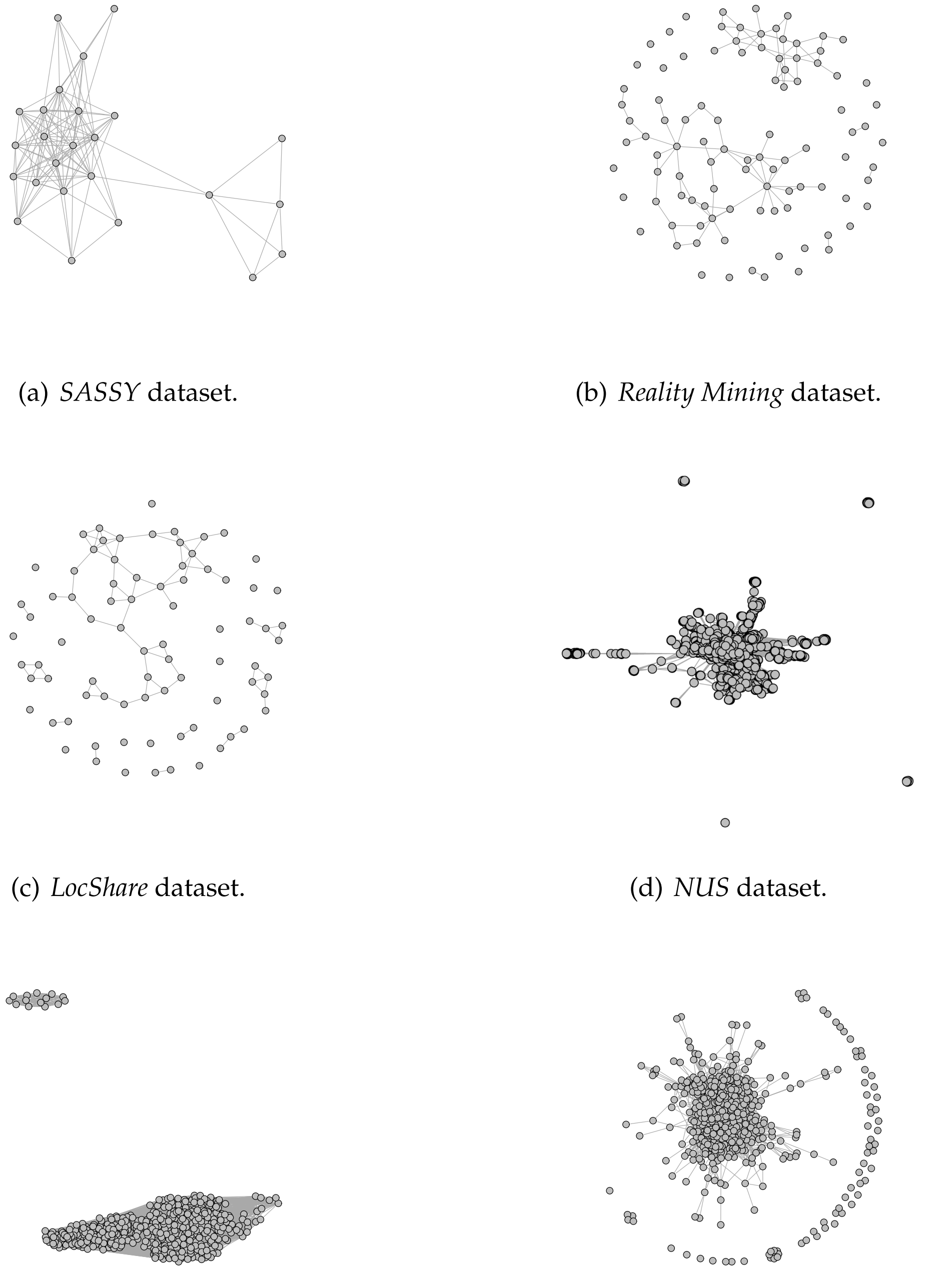
Dataset	Number of nodes		Clustering coefficient	Social links	Encounters
	Total	≥1 edge			
SASSY	25	25	0.771	254	29,909
Reality Mining	97	75	0.318	107	32,359
LocShare	80	64	0.470	81	2,400
NUS	22 341	22 340	0.536	6,261,458	12,320,946
NUS-L	500	500	0.634	29,743	71,819
NUS-R	500	476	0.506	3,001	6,109

Table A.1: Dataset statistics. All fields refer to properties of the social network, except for the number of encounters.

For *NUS-R*, the mean friends list size is 12.0, with a standard deviation of 8.54. For *NUS-L*, the mean friends list size is 119, with a standard deviation of 39.9. The wide difference in mean friends list size (while the absolute number of nodes is the same at 500) is an expected outcome from the different natures of the two *NUS* sampling methods, and provides an initial illustration of the differing properties of the derived datasets.

A.2 Dataset statistics

Table A.1 provides an overall description of the datasets used for evaluation; we have a chosen a variety of datasets which vary in size, timescale and density. This is further confirmed by visualisations of the various social networks within these datasets (Figure A.1) and the degree distributions of these social networks (Figure A.2), which indicate variety in the network structures. We therefore believe that these datasets provide a range of suitable test cases for opportunistic network performance simulations.



(a) *SASSY* dataset.

(b) *Reality Mining* dataset.

(c) *LocShare* dataset.

(d) *NUS* dataset.

(e) *NUS-L* dataset.

(f) *NUS-R* dataset.

Figure A.1: The social networks from the three datasets used for evaluation. The *NUS* dataset, Figure A.1(d), was sampled in two different ways to derive two different datasets — Figure A.1(e) and Figure A.1(f).

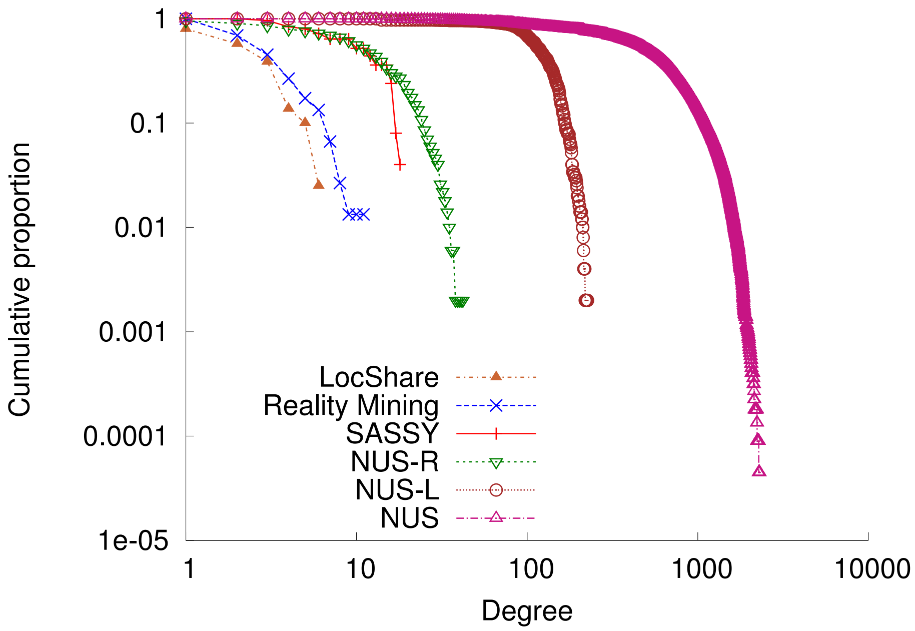


Figure A.2: Cumulative degree distributions for the datasets (log-log plot). Our datasets have a wide variance in their cumulative degree distributions.

Appendix B

Letter of ethical approval

A portion of the research included in this thesis required ethical approval from the University Teaching and Research Ethics Committee (UTREC):

- UTREC approval code: CS6278
- Title: Privacy Value Networks / Location-sharing with Mobile Phones on Facebook (LocShare)
- Experiment date: 2009–2010
- Researchers: Fehmi Ben Abdesslem, Iain Parris
- Supervisor: Tristan Henderson

The letter of ethical approval is included on the next page.

University of St Andrews

University Teaching and Research Ethics Committee

30/03/2010
Fehmi Ben Abdesslem, Iain Parris
School of Computer Science

Ethics Reference No: *Please quote this ref on all correspondence*	**CS6278**
Project Title:	**Privacy Value Networks**
Researchers Name(s):	**Fehmi Ben Abdesslem, Iain Parris**
Supervisor(s):	**Tristan Henderson**

Thank you for submitting your application which was considered at the Computer Science School Ethics Committee meeting on the 17th November 2009. The following documents were reviewed:

1. Ethical Application Form	17/11/09
2. Participant Information Sheet	17/11/09
3. Consent Form	17/11/09
~~4. Debriefing Form~~	
~~5. External Permissions~~	
~~6. Letters to Parents/Children/Headteacher etc…~~	
~~7. Questionnaires~~	
8. ~~Enhanced Disclosure Scotland and Equivalent~~ (*as necessary*)	

The University Teaching and Research Ethics Committee (UTREC) approves this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years. Projects, which have not commenced within two years of original approval, must be re-submitted to your School Ethics Committee.

You must inform your School Ethics Committee when the research has been completed. If you are unable to complete your research within the 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration, must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' (http://www.st-andrews.ac.uk/media/UTRECguidelines%20Feb%2008.pdf) are adhered to.

Yours sincerely

Convenor of the School Ethics Committee

Ccs Supervisor
School Ethics Committee

Appendix C

Data for Figure 5.1

Senders' friends list size change (target %)	OSNR false positive rate (%)				
	SASSY	*Reality Mining*	*LocShare*	*NUS-L*	*NUS-R*
-80	1.2×10^{-3}	2.0×10^{-2}	5.6×10^{-2}	6.5×10^{-3}	4.3×10^{-3}
-60	1.8×10^{-2}	7.7×10^{-2}	6.5×10^{-2}	8.7×10^{-2}	4.7×10^{-2}
-40	8.1×10^{-2}	2.2×10^{-1}	8.9×10^{-2}	3.5×10^{-1}	2.3×10^{-1}
-20	2.2×10^{-1}	6.5×10^{-1}	7.0×10^{-1}	9.4×10^{-1}	6.1×10^{-1}
0	4.7×10^{-1}	1.4×10^{0}	1.0×10^{0}	2.0×10^{0}	1.3×10^{0}
20	8.8×10^{-1}	1.8×10^{0}	1.4×10^{0}	3.4×10^{0}	2.4×10^{0}
40	1.4×10^{0}	4.5×10^{0}	2.9×10^{0}	5.3×10^{0}	3.7×10^{0}
60	2.2×10^{0}	5.2×10^{0}	3.8×10^{0}	7.6×10^{0}	5.4×10^{0}
80	2.8×10^{0}	8.0×10^{0}	7.7×10^{0}	1.0×10^{1}	7.8×10^{0}
100	3.2×10^{0}	1.0×10^{1}	8.2×10^{0}	1.4×10^{1}	9.8×10^{0}
120	3.7×10^{0}	1.2×10^{1}	8.1×10^{0}	1.7×10^{1}	1.3×10^{1}
140	4.1×10^{0}	1.7×10^{1}	1.5×10^{1}	2.1×10^{1}	1.6×10^{1}
160	4.3×10^{0}	1.7×10^{1}	1.8×10^{1}	2.4×10^{1}	1.9×10^{1}
180	4.7×10^{0}	2.3×10^{1}	2.1×10^{1}	2.8×10^{1}	2.3×10^{1}
200	4.9×10^{0}	2.6×10^{1}	2.2×10^{1}	3.1×10^{1}	2.6×10^{1}

Table C.1: *OSNR* false positive rate for each dataset (see Figure 5.1).

Bibliography

[1] I. Aad, C. Castelluccia, and J.-P. Huubaux. Packet coding for strong anonymity in ad hoc networks. In *2006 Securecomm and Workshops*, Aug. 2006. doi:10.1109/SECCOMW.2006.359571.

[2] N. Ahmad, H. Cruickshank, Z. Sun, and M. Asif. Pseudonymised communication in delay tolerant networks. In *Proceedings of the Ninth Annual International Conference on Privacy, Security and Trust (PST)*, July 2011. doi:10.1109/pst.2011.5971956.

[3] A. Al-Hinai, H. Zhang, Y. Chen, and Y. Li. TB-SnW: Trust-based spray-and-wait routing for delay-tolerant networks. *Journal of Supercomputing*, pages 1–17, 2014. doi:10.1007/s11227-014-1095-z.

[4] T. R. Andel and A. Yasinac. On the credibility of MANET simulations. *IEEE Computer*, 39(7):48–54, July 2006. doi:10.1109/MC.2006.242.

[5] E. Anderson, G. Yee, C. Phillips, D. Sicker, and D. Grunwald. The impact of directional antenna models on simulation accuracy. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on*, June 2009. doi:10.1109/wiopt.2009.5291643.

[6] D. Anthony, T. Henderson, and D. Kotz. Privacy in Location-Aware computing environments. *IEEE Pervasive Computing*, 6(4):64–72, Oct. 2007. doi:10.1109/MPRV.2007.83.

[7] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An Obfuscation-Based approach for protecting location privacy. *IEEE Transactions on Dependable and Secure Computing*, 8(1):13–27, Jan. 2011. doi:10.1109/tdsc.2009.25.

[8] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo. Applicability of identity-based cryptography for disruption-tolerant networking. In *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, MobiOpp '07, 2007. doi:10.1145/1247694.1247705.

[9] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, Aug. 2007. doi:10.1145/1282427.1282422.

[10] A. Barzan, B. Bonne, P. Quax, W. Lamotte, M. Versichele, and N. Van de Weghe. A comparative simulation of opportunistic routing protocols using realistic mobility data obtained from mass events. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, June 2013. doi:10.1109/wowmom.2013.6583438.

[11] S. K. Belle and M. Waldvogel. Consistent deniable lying: Privacy in mobile social networks. In *Proceedings of the Workshop on Security and Privacy Issues in Mobile Phone Use*, May 2008. Online at http://www.pervasive2008.org/Papers/Workshop/w1-03.pdf.

[12] F. Ben Abdesslem, T. Henderson, S. Brostoff, and M. A. Sasse. Context-based personalised settings for mobile location sharing. In *Proceedings of the ACM Recommender Systems Workshop on Personalization in Mobile Applications*, Oct. 2011.

[13] F. Ben Abdesslem, T. Henderson, and I. Parris. CRAWDAD data set st_andrews/locshare (v. 2011-10-12). Downloaded from http://crawdad.org/st_andrews/locshare/, Oct. 2011.

[14] F. Ben Abdesslem, I. Parris, and T. Henderson. Reliable online social network data collection. In A. Abraham and A. Ella Hassanien, editors, *Computational Social Networks: Mining and Visualization*, volume 3 of *Springer Computer Communications and Networks Series*. Springer-Verlag, London, UK, 2011. Accepted for publication.

[15] M. Benisch, P. Kelley, N. Sadeh, and L. Cranor. Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs. *Personal and*

Ubiquitous Computing, pages 1–16, Dec. 2010. doi:10.1007/s00779-010-0346-0.

[16] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, Apr. 2003. doi:10.1109/MPRV.2003.1186725.

[17] G. Bianchi, L. Bracciale, and P. Loreti. *"Better Than Nothing" Privacy with Bloom Filters: To What Extent?*, volume 7556 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-33627-0_27.

[18] G. Bigwood and T. Henderson. Bootstrapping opportunistic networks using social roles. In *Proceedings of the Fifth IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications*, June 2011. doi:10.1109/WoWMoM.2011.5986139.

[19] G. Bigwood and T. Henderson. IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. In *Proceedings of the IEEE Third International Conference on Social Computing (SocialCom)*, Oct. 2011. doi:10.1109/PASSAT/SocialCom.2011.60.

[20] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. Exploiting Self-Reported social networks for routing in ubiquitous computing environments. In *Proceedings of the 1st International Workshop on Social Aspects of Ubiquitous Computing Environments (SAUCE)*, Oct. 2008. doi:10.1109/WiMob.2008.86.

[21] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. CRAWDAD data set st_andrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.org/st_andrews/sassy, June 2011.

[22] D. Bittencourt, E. Mota, E. Nascimento Silva, and C. B. Souza. Towards realism in DTN performance evaluation using virtualization. In *Wireless Days (WD), 2013 IFIP*, Nov. 2013. doi:10.1109/wd.2013.6686511.

[23] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970. doi:10.1145/362686.362692.

[24] M. Boban and T. T. V. Vinhoza. *Modeling and Simulation of Vehicular Networks: towards Realistic and Efficient Models*. InTech, Jan. 2011. doi:10.5772/12846.

[25] C. Boldrini, M. Conti, and A. Passarella. Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. *Pervasive and Mobile Computing*, 4(5):633–657, Oct. 2008. doi:10.1016/j.pmcj.2008.04.003.

[26] D. Boneh and M. Franklin. Identity-Based encryption from the weil pairing. *SIAM Journal on Computing*, 32(3):586–615, Jan. 2003. doi:10.1137/s0097539701398521.

[27] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, Apr. 1998. doi:10.1016/s0169-7552(98)00110-x.

[28] A. J. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Ubicomp '10: Proceedings of the 12th ACM international conference on Ubiquitous computing*, Sept. 2010. doi:10.1145/1864349.1864381.

[29] E. Bulut and B. K. Szymanski. Secure multi-copy routing in compromised delay tolerant networks. *Wireless Personal Communications*, 73(1):149–168, Nov. 2013. doi:10.1007/s11277-012-0960-4.

[30] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine. Surviving attacks on disruption-tolerant networks without authentication. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, Sept. 2007. doi:10.1145/1288107.1288116.

[31] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, Apr. 2006. doi:10.1109/infocom.2006.228.

[32] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, June 2003. doi:10.1109/MCOM.2003.1204759.

[33] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002. doi:10.1002/wcm.72.

[34] V. G. Cerf, S. C. Burleigh, A. J. Hooke, L. Torgerson, R. C. Durst, K. L. Scott, K. Fall, and H. S. Weiss. Delay-Tolerant networking architecture. RFC 4838 (Informational), Apr. 2007. Online at https://tools.ietf.org/rfc/rfc4838.txt.

[35] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, June 2007. doi:10.1109/tmc.2007.1060.

[36] L.-J. Chen, C.-L. Chiou, and Y.-C. Chen. An evaluation of routing reliability in non-collaborative opportunistic networks. In *2009 International Conference on Advanced Information Networking and Applications (AINA)*, May 2009. doi:10.1109/AINA.2009.54.

[37] F. C. Choo, M. C. Chan, and E.-C. Chang. Robustness of DTN against routing attacks. In *2010 Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010)*, Jan. 2010. doi:10.1109/COMSNETS.2010.5432014.

[38] M. Chuah, P. Yang, and J. Han. A ferry-based intrusion detection scheme for sparsely connected ad hoc networks. In *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, 2007. doi:10.1109/mobiq.2007.4451068.

[39] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626 (Experimental), Oct. 2003. Online at http://tools.ietf.org/pdf/rfc3626.pdf.

[40] S. Consolvo, I. E. Smith, T. Matthews, A. Lamarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, Apr. 2005. doi:10.1145/1054972.1054985.

[41] S. Consolvo and M. Walker. Using the experience sampling method to evaluate ubicomp applications. *Pervasive Computing, IEEE*, 2(2):24–31, 2003. doi:10.1109/MPRV.2003.1203750.

[42] M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, Jan. 2010. doi:10.1109/MC.2010.19.

[43] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748–760, June 2008. doi:10.1109/jsac.2008.080602.

[44] G. Costantino, F. Martinelli, and P. Santi. Investigating the privacy versus forwarding accuracy tradeoff in opportunistic Interest-Casting. *IEEE Transactions on Mobile Computing*, 13(4), Apr. 2013. doi:10.1109/tmc.2013.20.

[45] J. Crowcroft, E. Yoneki, P. Hui, and T. Henderson. Promoting tolerance for delay tolerant network research. *SIGCOMM Comput. Commun. Rev.*, 38(5):63–68, 2008. doi:10.1145/1452335.1452345.

[46] M. Csikszentmihalyi and R. Larson. Validity and reliability of the Experience-Sampling method. *J Nerv Ment Dis*, 175(9):526–536, Sept. 1987. Online at http://view.ncbi.nlm.nih.gov/pubmed/3655778.

[47] E. M. Daly and M. Haahr. Social network analysis for information flow in disconnected delay-tolerant MANETs. *IEEE Transactions on Mobile Computing*, 8(5):606–621, May 2009. doi:10.1109/TMC.2008.161.

[48] M. L. Damiani, C. Silvestri, and E. Bertino. Fine-Grained cloaking of sensitive positions in Location-Sharing applications. *IEEE Pervasive Computing*, 10(4):64–72, Apr. 2011. doi:10.1109/mprv.2011.18.

[49] H. Deng, W. Li, and D. P. Agrawal. Routing security in wireless ad hoc networks. *Communications Magazine, IEEE*, 40(10):70–75, Oct. 2002. doi:10.1109/mcom.2002.1039859.

[50] L. Dóra and T. Holczer. Hide-and-Lie: enhancing application-level privacy in opportunistic networks. In *Proceedings of the 2nd International Workshop on Mobile Opportunistic Networking (MobiOpp)*, Feb. 2010. doi:10.1145/1755743.1755767.

[51] J. R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, 2002. doi:10.1007/3-540-45748-8_24.

[52] N. Eagle and A. S. Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from http://crawdad.cs.dartmouth.edu/mit/reality, July 2005.

[53] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, Aug. 2009. doi:10.1073/pnas.0900282106.

[54] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models (MobilityModels)*, 2008. doi:10.1145/1374688.1374695.

[55] K. El Defrawy, J. Solis, and G. Tsudik. Leveraging social contacts for message confidentiality in delay tolerant networks. In *2009 33rd Annual IEEE International Computer Software and Applications Conference*, July 2009. doi:10.1109/COMPSAC.2009.43.

[56] K. El Defrawy and G. Tsudik. Privacy-preserving location-based on-demand routing in MANETs. *IEEE Journal on Selected Areas in Communications*, 29(10):1926–1934, Dec. 2011. doi:10.1109/jsac.2011.111203.

[57] K. Fall. A Delay-Tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, Aug. 2003. doi:10.1145/863955.863960.

[58] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, Aug. 2001. doi:10.1109/90.944338.

[59] R. Friedman, M. Gradinariu, and G. Simon. Locating cache proxies in MANETs. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, May 2004. doi:10.1145/989459.989482.

[60] J. Froehlich, M. Chen, I. Smith, and F. Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. In P. Dourish and A. Friday, editors, *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006. doi:10.1007/11853565_20.

[61] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and I will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, Nov. 2010. doi:10.1145/1868470.1868479.

[62] K. Garg, A. Förster, D. Puccinelli, and S. Giordano. *Towards Realistic and Credible Wireless Sensor Network Evaluation*, volume 89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-29096-1_4.

[63] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazières, and H. Yu. RE: reliable email. In *NSDI 2006: 3rd Symposium on Networked Systems Design & Implementation*. USENIX Association. Online at http://www.usenix.org/events/nsdi06/tech/full_papers/garriss/garriss.pdf.

[64] D. Goldschlag, M. Reed, and P. Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, Feb. 1999. doi:10.1145/293411.293443.

[65] L. Guo, C. Zhang, H. Yue, and Y. Fang. PSaD: A privacy-preserving social-assisted content dissemination scheme in DTNs. *IEEE Transactions on Mobile Computing*, page 1, 2014. doi:10.1109/tmc.2014.2308177.

[66] Y. Guo, S. Gordon, and S. Perreau. A flow based detection mechanism against flooding attacks in mobile ad hoc networks. In *2007 IEEE Wireless Communications and Networking Conference*, 2007. doi:10.1109/WCNC.2007.574.

[67] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, volume 2, Oct. 1997. doi:10.1109/icupc.1997.627227.

[68] S. Hameed, X. Fu, P. Hui, and N. Sastry. LENS: Leveraging social networking and trust to prevent spam transmission. In *Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP)*, Oct. 2011. doi:10.1109/icnp.2011.6089044.

[69] O. Hasan, J. Miao, S. Ben Mokhtar, and L. Brunie. A privacy preserving prediction-based routing protocol for mobile delay tolerant networks. In

2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Mar. 2013. doi:10.1109/AINA.2013.6.

[70] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical report, 2007. Online at http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1175;context=cs_faculty_pubs.

[71] D. Hiranandani, K. Obraczka, and J. J. Garcia-Luna-Aceves. MANET protocol simulations considered harmful: the case for benchmarking. *Wireless Communications, IEEE*, 20(4), Aug. 2013. doi:10.1109/mwc.2013.6590054.

[72] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *Network, IEEE*, 16(4):11–21, July 2002. doi:10.1109/mnet.2002.1020231.

[73] D. Huff. *How to Lie with Statistics*. W. W. Norton & Company, reissue edition, 1954.

[74] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, Aug. 2005. doi:10.1145/1080139.1080142.

[75] P. Hui, J. Crowcroft, and E. 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 '08, May 2008. doi:10.1145/1374618.1374652.

[76] R. Jain, D. Lelescu, and M. Balakrishnan. Model T: an empirical model for user registration patterns in a campus wireless LAN. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, Aug. 2005. doi:10.1145/1080829.1080848.

[77] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.*, 34(4):145–158, Aug. 2004. doi:10.1145/1015467.1015484.

[78] R. Jansen and R. Beverly. Toward anonymity in delay tolerant networks: Threshold pivot scheme. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, Oct. 2010. doi:10.1109/milcom.2010.5680442.

[79] S. Joerer, C. Sommer, and F. Dressler. Toward reproducibility and comparability of IVC simulation studies: a literature survey. *Communications Magazine, IEEE*, 50(10):82–88, Oct. 2012. doi:10.1109/mcom.2012.6316780.

[80] D. Johnson, Y.-C. Hu, and D. Maltz. The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. RFC 4728 (Experimental), Feb. 2007. Online at http://tools.ietf.org/pdf/rfc4728.pdf.

[81] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. F. Korth, editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*. Springer US, Boston, MA, USA, 1996. doi:10.1007/978-0-585-29603-6_5.

[82] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, Oct. 2002. doi:10.1145/635508.605408.

[83] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing Source-Location privacy in sensor network routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, June 2005. doi:10.1109/ICDCS.2005.31.

[84] A. Kapadia, T. Henderson, J. Fielding, and D. Kotz. Virtual walls: Protecting digital privacy in pervasive environments. In *Proceedings of the 5th International Conference on Pervasive Computing*, number 4480 in LNCS, May 2007. doi:10.1007/978-3-540-72037-9_10.

[85] A. Kate, G. M. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *Security and Privacy in Communications Networks*

and the Workshops, 2007. SecureComm 2007. Third International Conference on, 2007. doi:10.1109/seccom.2007.4550373.

[86] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools)*, Mar. 2009. doi:10.4108/ICST.SIMUTOOLS2009.5674.

[87] Y. Kim and A. Helmy. CATCH: A protocol framework for cross-layer attacker traceback in mobile multi-hop networks. *Ad Hoc Networks*, 8(2):193–213, Mar. 2010. doi:10.1016/j.adhoc.2009.07.002.

[88] Y. Kim, K. Taylor, C. Dunbar, B. Walker, and P. Mundur. Reality vs emulation: running real mobility traces on a mobile wireless testbed. In *Proceedings of the 3rd ACM international workshop on Hot Topics in Planet-Scale Measurement*, 2011. doi:10.1145/2000172.2000180.

[89] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *2008 IEEE 24th International Conference on Data Engineering (ICDE 2008)*, Apr. 2008. doi:10.1109/icde.2008.4497554.

[90] J. Krumm. Realistic driving trips for location privacy. In H. Tokuda, M. Beigl, A. Friday, A. Brush, and Y. Tobe, editors, *Pervasive Computing*, volume 5538 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-01516-8_4.

[91] S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: the incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(4):50–61, Oct. 2005. doi:10.1145/1096166.1096174.

[92] Z. Le, G. Vakde, and M. Wright. PEON: privacy-enhanced opportunistic networks with applications in assistive environments. In *PETRA '09: Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments*, June 2009. doi:10.1145/1579114.1579190.

[93] F. C. Lee, W. Goh, and C. K. Yeo. A queuing mechanism to alleviate flooding attacks in probabilistic delay tolerant networks. In *2010 Sixth Advanced International Conference on Telecommunications*, 2010. doi:10.1109/AICT.2010.78.

[94] F. Li, J. Wu, and A. Srinivasan. Thwarting blackhole attacks in Disruption-Tolerant networks using encounter tickets. In *INFOCOM 2009, IEEE*, Apr. 2009. doi:10.1109/infcom.2009.5062170.

[95] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. In P. Dini, P. Lorenz, and J. e. u. m. a. n. Souza, editors, *Proceedings of the First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR)*, volume 3126 of *Lecture Notes in Computer Science*, Aug. 2004. doi:10.1007/978-3-540-27767-5_24.

[96] C. Liu and J. Wu. Routing in a cyclic mobispace. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, 2008. doi:10.1145/1374618.1374665.

[97] R. Lu, X. Lin, and X. Shen. SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *Proceedings of IEEE INFOCOM 2010*, Mar. 2010. doi:10.1109/INFCOM.2010.5462161.

[98] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong. Anti-localization anonymous routing for delay tolerant network. *Computer Networks*, 54(11):1899–1910, Aug. 2010. doi:10.1016/j.comnet.2010.03.002.

[99] A. J. Mashhadi, S. Ben Mokhtar, and L. Capra. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In *Proceedings of the 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2009. doi:10.1109/WOWMOM.2009.5282467.

[100] G. Mezzour, A. Perrig, V. Gligor, and P. Papadimitratos. Privacy-Preserving relationship path discovery in social networks. In J. A. Garay, A. Miyaji, and A. Otsuka, editors, *Cryptology and Network Security*, volume 5888. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-10433-6_13.

[101] J. Miao, O. Hasan, S. Ben Mokhtar, L. Brunie, and K. Yim. An investigation on the unwillingness of nodes to participate in mobile delay tolerant

network routing. *International Journal of Information Management*, 33(2):252–262, Apr. 2013. doi:10.1016/j.ijinfomgt.2012.11.001.

[102] S. Milgram. The familiar stranger: An aspect of urban anonymity. In *The Individual in a Social World*. Addison-Wesley, Reading, MA, USA, Aug. 1977.

[103] V. F. S. Mota, F. D. Cunha, D. F. Macedo, J. M. S. Nogueira, and A. A. F. Loureiro. Protocols, mobility models and tools in opportunistic networks: A survey. *Computer Communications*, Mar. 2014. doi:10.1016/j.comcom.2014.03.019.

[104] A. Mtibaa, M. May, C. Diot, and M. Ammar. PeopleRank: Social opportunistic forwarding. In *Proceedings of IEEE INFOCOM 2010*, Mar. 2010. doi:10.1109/INFCOM.2010.5462261.

[105] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, volume 0, May 2009. doi:10.1109/SP.2009.22.

[106] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott. Experimental evaluation of wireless simulation assumptions. *SIMULATION*, 83(9):643–661, Sept. 2007. doi:10.1177/0037549707085632.

[107] P. Oechslin. Making a faster cryptanalytic Time-Memory Trade-Off. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*. Springer, 2003. doi:10.1007/978-3-540-45146-4_36.

[108] G. K. Orman and V. Labatut. The effect of network realism on community detection algorithms. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, Aug. 2010. doi:10.1109/ASONAM.2010.70.

[109] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, Jan. 2002. Online at http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-35556.

[110] I. Parris, G. Bigwood, and T. Henderson. Privacy-enhanced social network routing in opportunistic networks. In *2010 8th IEEE International Conference*

on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mar. 2010. doi:10.1109/PERCOMW.2010.5470511.

[111] I. Parris and T. Henderson. Privacy-enhanced social-network routing. *Computer Communications*, 35(1):62–74, Jan. 2012. doi:10.1016/j.comcom.2010.11.003.

[112] K. Pawlikowski, H. D. J. Jeong, and J. S. R. Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 40(1):132–139, Jan. 2002. doi:10.1109/35.978060.

[113] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *IEEE Commun. Surveys Tuts.*, 13(2):245–257, 2011. doi:10.1109/surv.2011.041110.00022.

[114] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, Nov. 2006. doi:10.1109/MCOM.2006.248176.

[115] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand distance vector (AODV) routing. RFC 3561 (Experimental), July 2003. Online at http://tools.ietf.org/pdf/rfc3561.pdf.

[116] L. F. Perrone, C. S. Main, and B. C. Ward. SAFE: Simulation automation framework for experiments. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, Dec. 2012. doi:10.1109/wsc.2012.6465286.

[117] C. Piro, C. Shields, and B. N. Levine. Detecting the sybil attack in mobile ad hoc networks. In *2006 Securecomm and Workshops*, Aug. 2006. doi:10.1109/SECCOMW.2006.359558.

[118] M. Radenkovic, I. Vaghi, S. Zakhary, and A. Benslimane. AdaptAnon: Adaptive anonymity for service queries in mobile opportunistic networks. In *Communications (ICC), 2013 IEEE International Conference on*, June 2013. doi:10.1109/icc.2013.6654788.

[119] D. P. Reed. That sneaky exponential - beyond Metcalfe's law to the power of community building. *Context Magazine*, Spring 1999.

[120] N. Ristanovic, G. Theodorakopoulos, and J. Y. Le Boudec. Traps and pitfalls of using contact traces in performance studies of opportunistic networks. In *INFOCOM, 2012 Proceedings IEEE*, Mar. 2012. doi:10.1109/infcom.2012.6195502.

[121] N. Ristanovic, D. K. Tran, and J. Y. Le Boudec. Tracking of mobile devices through bluetooth contacts. In *Proceedings of the ACM CoNEXT Student Workshop*, CoNEXT '10 Student Workshop, 2010. doi:10.1145/1921206.1921211.

[122] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. Wiley, 1 edition, Jan. 2004. Online at http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471453803.

[123] R. Schnell, T. Bachteler, and J. Reiher. Privacy-preserving record linkage using Bloom filters. *BMC Medical Informatics and Decision Making*, 9(1):41+, Aug. 2009. doi:10.1186/1472-6947-9-41.

[124] M. Schwamborn, N. Aschenbruck, and P. Martini. A realistic trace-based mobility model for first responder scenarios. In *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, MSWIM '10, 2010. doi:10.1145/1868521.1868564.

[125] A. Seth and S. Keshav. Practical security for disconnected nodes. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005. (NPSec).*, Nov. 2005. doi:10.1109/npsec.2005.1532050.

[126] C. Shi, X. Luo, P. Traynor, M. H. Ammar, and E. W. Zegura. ARDEN: Anonymous networking in delay tolerant networks. *Ad Hoc Networks*, 10(6):918–930, Aug. 2012. doi:10.1016/j.adhoc.2011.11.008.

[127] A. Shikfa, M. Önen, and R. Molva. Privacy and confidentiality in context-based and epidemic forwarding. *Computer Communications*, 33(13):1493–1504, Apr. 2010. doi:10.1016/j.comcom.2010.04.035.

[128] A. Socievole, F. De Rango, and S. Marano. Face-to-face with facebook friends: Using online friendlists for routing in opportunistic networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, 2013. doi:10.1109/pimrc.2013.6666659.

[129] A. Socievole, F. De Rango, and S. Marano. Link prediction in human contact networks using online social ties. In *Cloud and Green Computing (CGC), 2013 Third International Conference on*, 2013. doi:10.1109/cgc.2013.55.

[130] L. Song and D. F. Kotz. Evaluating opportunistic routing protocols with large realistic contact traces. In *CHANTS '07: Proceedings of the second ACM workshop on Challenged networks*, Aug. 2007. doi:10.1145/1287791.1287799.

[131] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, 2005. doi:10.1145/1080139.1080143.

[132] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proceedings of the Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom)*, Sept. 2006. doi:10.1145/1161089.1161100.

[133] V. Srinivasan, M. Motani, and W. T. Ooi. CRAWDAD data set nus/contact (v. 2006-08-01). Downloaded from http://crawdad.cs.dartmouth.edu/nus/contact, Aug. 2006.

[134] J. Su, A. Goel, and E. de Lara. An empirical evaluation of the Student-Net delay tolerant network. In *Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on*, 2006. doi:10.1109/mobiq.2006.340403.

[135] J. Su, J. Scott, P. Hui, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. 2007. doi:10.1007/978-3-540-74853-3_23.

[136] K. Tan, D. Wu, A. Jack Chan, and P. Mohapatra. Comparing simulation tools and experimental testbeds for wireless mesh networks. In *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2010. doi:10.1109/WOWMOM.2010.5534917.

[137] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh. Empirical models of privacy in location

sharing. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, Sept. 2010. doi:10.1145/1864349.1864364.

[138] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre. Preventing spam in opportunistic networks. *Computer Communications*, 41:31–42, Mar. 2014. doi:10.1016/j.comcom.2013.12.003.

[139] S. Trifunovic, F. Legendre, and C. Anastasiades. Social trust in opportunistic networks. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*, Mar. 2010. doi:10.1109/INFCOMW.2010.5466696.

[140] M. Y. Uddin, A. Khurshid, H. D. Jung, C. Gunter, M. Caesar, and T. Abdelzaher. Making DTNs robust against spoofing attacks with localized countermeasures. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2011. doi:10.1109/sahcn.2011.5984915.

[141] A. Vahdat and D. Becker. Epidemic routing for Partially-Connected ad hoc networks. Technical report, Duke University, Apr. 2000. Online at http://issg.cs.duke.edu/epidemic/epidemic.pdf.

[142] C. Vallati, V. Omwando, and P. Mohapatra. *Experimental Work Versus Simulation in the Study of Mobile Ad Hoc Networks*. John Wiley & Sons, Inc., Hoboken, NJ, USA, second edition, Feb. 2013. doi:10.1002/9781118511305.ch6.

[143] A. F. Westin. Social and political dimensions of privacy. *Journal of Social Issues*, 59(2):431–453, July 2003. doi:10.1111/1540-4560.00072.

[144] A. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, Oct. 2002. doi:10.1109/mc.2002.1039518.

[145] K. Xu, P. Hui, V. O. Li, J. Crowcroft, V. Latora, and P. Lio. Impact of altruism on opportunistic communications. In *Proceedings of the First International Conference on Ubiquitous and Future Networks (ICUFN)*, June 2009. doi:10.1109/ICUFN.2009.5174303.

[146] M. Xue, P. Kalnis, and H. Pung. Location diversity: Enhanced privacy protection in location based services. In T. Choudhury, A. Quigley,

T. Strang, and K. Suginuma, editors, *Location and Context Awareness*, volume 5561 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-01721-6_5.

[147] P. Yi, Z. Dai, Y.-P. Zhong, and S. Zhang. Resisting flooding attacks in ad hoc networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, Apr. 2005. doi:10.1109/itcc.2005.248.

[148] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, Apr. 2003. doi:10.1109/infcom.2003.1208967.

[149] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, 2006. doi:10.1145/1159913.1159945.

[150] S. Zakhary and M. Radenkovic. Erasure coding with replication to defend against malicious attacks in DTN. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, Oct. 2011. doi:10.1109/wimob.2011.6085375.

[151] S. Zakhary and M. Radenkovic. Utilizing social links for location privacy in opportunistic delay-tolerant networks. In *Communications (ICC), 2012 IEEE International Conference on*, June 2012. doi:10.1109/icc.2012.6364413.

[152] S. Zakhary, M. Radenkovic, and A. Benslimane. The quest for location-privacy in opportunistic mobile social networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, July 2013. doi:10.1109/iwcmc.2013.6583637.

[153] S. Zakhary, M. Radenkovic, and A. Benslimane. Efficient location Privacy-Aware forwarding in opportunistic mobile networks. *IEEE Transactions on Vehicular Technology*, 63(2):893–906, Feb. 2014. doi:10.1109/tvt.2013.2279671.

[154] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc 2004: Proceedings of*

the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, May 2004. doi:10.1145/989459.989483.