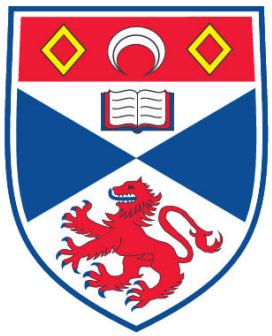
USING SELF-REPORTED SOCIAL NETWORKS TO IMPROVE OPPORTUNISTIC NETWORKING

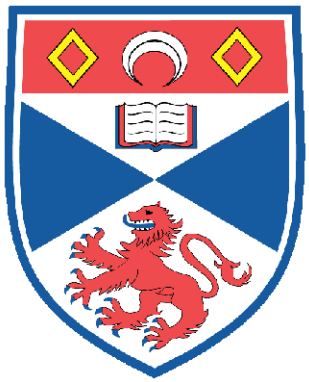
Greg Bigwood

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



2012

UNIVERSITY OF ST ANDREWS
SCHOOL OF COMPUTER SCIENCE



Using Self-Reported Social Networks to Improve Opportunistic Networking

A THESIS TO BE SUBMITTED TO

THE UNIVERSITY OF ST ANDREWS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Author:

Greg Bigwood

email:gjb4@st-andrews.ac.uk

Supervisor:

Dr Tristan Henderson

email:tnhh@st-andrews.ac.uk

15th January 2012

Abstract

Opportunistic networks provide an ad hoc communication medium without the need for an infrastructure network, by leveraging human encounters and mobile devices. Routing protocols in opportunistic networks frequently rely upon encounter histories to build up meaningful data to use for informed routing decisions. This thesis shows that it is possible to use pre-existing social-network information to improve existing opportunistic routing protocols, and that these *self-reported social networks* have a particular benefit when used to bootstrap an opportunistic routing protocol.

Frequently, opportunistic routing protocols require users to relay messages on behalf of one another: an act that incurs a cost to the relaying node. Nodes may wish to avoid this forwarding cost by not relaying messages. Opportunistic networks need to incentivise participation and discourage the selfish behaviour. This thesis further presents an incentive mechanism that uses *self-reported social networks* to construct and maintain reputation and trust relationships between participants, and demonstrates its superior performance over existing incentive mechanisms.

Declaration

I, Gregory Bigwood, hereby certify that this thesis, which is approximately 29000 words in length, has been written by me, that it is the record of work carried out by me, and that it has not been submitted in any previous application for a higher degree.

date _____ *signature of candidate* _____

I was admitted as a research student in October 2007 and as a candidate for the degree of Doctor of Philosophy in October 2007; the higher study of which this is a record was carried out in the University of St Andrews between October 2007 and September 2011.

date _____ *signature of candidate* _____

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* _____

In submitting this thesis to the University of St Andrews we understand that we are 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. We 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. We 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 electronic publication of this thesis:

Access to Printed copy and electronic publication of thesis through the University of St Andrews.

signature of candidate _____ *signature of supervisor* _____

date _____

"Great things are not done by impulse, but by a series of small things brought together." –Vincent Van Gogh

Contents

List of Figures

List of Tables

Acknowledgement

Thanks to Tristan for help and guidance throughout my studies.

Thanks to my parents for their love and support.

Thanks to all my friends in St Andrews for keeping me focused, in particular Angus and Devan.

Thanks to all my friends from home for keeping me grounded.

Thanks to Max for getting me into shape physically.

Thanks to Tamsin for proof reading and getting me into shape mentally.

Most importantly: thanks for reading.

Chapter 1

Introduction

The use of infrastructure data networks from mobile devices is growing and has tripled every year since 2007. [1] People are carrying a mobile phone at almost all times, and a large percentage of these devices tend to be smartphones. [2] Users can install a wide range of applications on these devices, providing more functionality over traditional mobile phones, which are limited to messaging and voice calls. Smartphones have rechargeable batteries meaning users can take the device anywhere and charge it when a suitable electrical connection is available.

Many personal communication devices allow connection to existing infrastructure networks, for example, laptops. There are, however, certain situations in which existing mobile or wireless (802.11 based or otherwise) networks are not adequate/appropriate.

- There are times when wireless signal is insufficient, or connectivity is unobtainable and therefore mobile devices are unable to connect to a network. This could be due to lack of infrastructure or too great a distance from the necessary network device.
- Batteries frequently run out of energy, therefore the wireless radio component may not be permanently activated. In networks without infrastructure connectivity, this can lead to partitions of the network and destruction of forwarding paths.

[1] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[2] http://blog.nielsen.com/nielsenwire/consumer/smartphones-to-overtake-feature-phones-in-u-s-by-2011/

In a world where we are becoming increasingly reliant on mobile communication in all aspects of our lives, the inability to communicate in such a manner can negatively affect business and personal relationships. In regions where there is no suitable network infrastructure, we need an alternative system that is present where the people are, regardless of the availability of fixed infrastructure.

For situations where existing IP and GSM/UMTS networks are unsuitable, researchers have developed alternative protocols to deal with these scenarios, specifically the study of so called "Delay Tolerant Networks" [48] (DTNs). A DTN works in situations where existing network protocols are unsuitable. DTNs use a store-and-forward architecture to allow communication when a path through the network is not reliable (due to likelihood of disconnection). Nodes may store duplications of messages for other nodes and then forward these on at the appropriate time.

Another way for a network to provide coverage for locations where people are is via an opportunistic network [115]. Opportunistic networks are a special case of delay-tolerant networks in which there are frequent disconnections and unpredictable encounter patterns; nodes forward messages when the opportunity arises: during an encounter contact.

The most basic routing protocol employed by these networks is Epidemic Routing [142] (named after disease spreading in epidemiology), whereby nodes flood the links by forwarding all stored messages at every encounter. This approach is effective at getting messages to their destination, however, it can rapidly drain the battery of the device, as there are many duplications of the messages.

To combat this energy drain, researchers have developed alternative routing protocols that reduce message duplications by restricting which messages nodes forward during an encounter. Different protocols use different criteria to reduce the number of message duplications. Many of these protocols use an encounter history to provide data used to determine the restrictions. An encounter-history–based protocol, however, takes time to build up a knowledge database from which to make accurate routing predictions about the network state. Thus we are left with the question:

Q1 How can we bootstrap an opportunistic network?

Moreover, we must consider the issue of incentives for participating in an opportunistic network. Forwarding messages incurs an energy cost. A rational approach for nodes, therefore, would be to not forward messages on behalf of other nodes, thereby allowing them to preserve their own battery power. We are thus confronted with the following question:

Q2 How can we incentivise participation in opportunistic DTNs?

If we are not going to use the encounter history to distinguish nodes for forwarding, we need an alternative approach for deciding when to forward messages. Since opportunistic networks frequently involve the mobility of humans, these human interaction networks frequently fall into the realm of small-world or "social networks", a topic covered extensively in the social sciences. In recent years we have seen the growth of online social networks, networks which may be useful for improving opportunistic networks, if they provide us with insights into user behaviour. These online social networks can provide us with "self-reported social networks", where nodes in the network provide a list of nodes they consider themselves to have a connection with: we consider these connections as ties in the network. An area of interest here is the division of the structure of the network into social roles. Using knowledge of the structure of the network we can divide the nodes into different roles depending on their behaviour.

1.1 Thesis

I have proposed that existing opportunistic routing protocols rely on encounter history and so under-perform. I therefore offer the following thesis:

> Self-reported social networks provide an alternative to encounter histories for efficient routing in opportunistic networks.

To address this statement I make three research contributions:

1. I show that self-reported social networks can be used for efficient message forwarding.

2. I show that existing history-based protocols do not provide adequate performance during the initial part of the network operation. I address this by providing a routing protocol using social roles that performs well when existing protocols cannot: when bootstrapping.

3. I demonstrate the use of self-reported social networks for developing a mechanism to detect selfishness by nodes and incentivise routing protocol co-operation.

1.2 Goals And Approach

This thesis attempts to demonstrate that the use of self-reported social network information in opportunistic forwarding can provide performance improvements to existing networks. The aim is to answer the questions 1-3 listed above. To do so, the primary approach is to construct a role-based forwarding protocol and compare the performance against existing protocols. As it is costly to procure many devices and users for an expansive test, we carry out the tests via trace-driven simulation, with a variety of representative scenarios and routing protocols.

To examine Q1, we analyse the performance of an alternative to using encounter histories. This demonstrates that an encounter history is not necessary for comparably efficient forwarding in an opportunistic network.

To examine Q2, we compare a mechanism for detecting selfishness using self-reported social networking against similar protocols. We see that self-reported social network information is useful in this scenario, and that we can use it to provide incentives for user co-operation.

1.3 Dissertation Outline

Chapters 2 and 3 provide an overview of the related literature and the current state of the art in the area.

- Chapter 2 describes the background and related research for opportunistic networking. The chapter shows how social networks can be used for opportunistic routing, and describes attacks against opportunistic networks

- Chapter 3 shows the current state of the art and highlights the problems that we address in this thesis

Chapters 4, 5 and 6 document the analysis and work performed to establish this thesis, and represent my contribution.

- Chapter 4 demonstrates that encounter histories are not necessary for adequate message passing, and that an alternative method provides similar performance

- Chapter 5 details the role-based forwarding protocol developed as to demonstrate social roles as an alternative to existing history based forwarding protocols. This chapter demonstrates the benefits of the new protocol, particularly for bootstrapping networks

- Chapter 6 details how we can use self-reported social network information to develop a distributed mechanism to detect selfish behaviour by nodes, and its performance compared to similar mechanisms. Then we show that we can incentivise co-operation by punishing selfish behaviour of nodes.

Finally, Chapter 7 concludes with a summary of the contributions of this thesis, as well as discussing the potential future research that can stem from this work.

1.4 Publications

During the course of my studies I have published several peer-reviewed works. The reader should note that while the major contributions in the design, analysis and implementation of these works were my own, I acknowledge the contribution and guidance of my fellow authors.

Exploiting self-reported social networks for routing in ubiquitous computing environments.
In Proceedings of IEEE SAUCE, Avignon, France, October 2008.
Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti.

Social DTN routing. (Poster Paper)
In Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT '08, New York, NY, USA.
Greg Bigwood and Tristan Henderson.

Social Delay-Tolerant network routing.
In Proceedings of the 2nd ICC Winter Workshop on Complexity in Social Systems, Lisbon, January 2009.
Greg Bigwood, Tristan Henderson, and Saleem Bhatti.

Social roles for opportunistic forwarding. (Poster Paper)
In Proceedings of the Second International Workshop on Mobile Opportunistic Networking, MobiOpp '10, pages 199–200, New York, NY, USA, February 2010. ACM.
Greg Bigwood and Tristan Henderson.

Bootstrapping Opportunistic Networks Using Social Roles.
In Proceedings of the Fifth IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2011), Lucca, Italy, June 2011.
Greg Bigwood and Tristan Henderson.

Collaborative data collection in global sensing systems.
In Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN 2011), Bonn, Germany, October 2011.

Greg Bigwood, Aline Carneiro Viana, Marcelo Dias de Amorim, and Mathias Boc.

IRONMAN: Using social networks to add incentives and reputation to opportunistic networks.
In Proceedings of the 3rd IEEE Conference on Social Computing (SocialCom 2011) MIT, Boston, USA, October 2011.
Greg Bigwood and Tristan Henderson.

During my studies I was involved with the following peer-reviewed works, where the main contributions were the work of co-authors.

Revisiting inter-flow fairness.
In Proceeding of the 5th International Conference on Broadband Communications, Networks and Systems, pages 585âĂŞ592. IEEE, September 2008.
Saleem Bhatti, Martin Bateman, Dimitrios Miras, Greg Bigwood, and Devan Rehunathan.

Privacy-enhanced social network routing in opportunistic networks.
In Proceedings of the Second IEEE International Workshop on SEcurity and SOCial Networking (SESOC), pages 624âĂŞ629, Mannheim, Germany, March 2010. IEEE Computer Society Press.
Iain Parris, Greg Bigwood, and Tristan Henderson.

Chapter 2

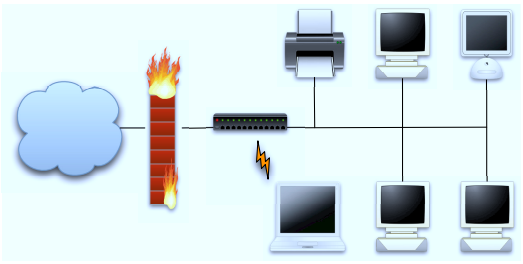
The development of opportunistic networks, and the usefulness of social networks

In this chapter we discuss using opportunistic networks in scenarios for which existing networking technologies are inappropriate. We see that we can use social network information to aid opportunistic networking.

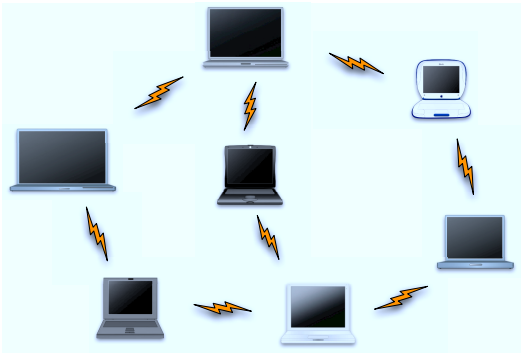
First, we see why delay in certain scenarios renders existing technology inappropriate for those scenarios. Second, we discuss the development of networks designed to cope with high-delay environments. Third, we discuss the similarities between social and mobile networks, and how we can use information gathered about social networks to improve understanding of mobile and opportunistic networks. Lastly, we look at attacks against opportunistic network routing.

We start by looking at wireless networking, which is important because mobile devices are becoming more prevalent, and as a result we are becoming increasingly reliant on communication whilst mobile.

Wired infrastructure networks are suitable for scenarios where nodes do not need to change location frequently. Examples of suitable scenarios include offices with a fixed



(a) An example small network. Black lines indicate wired connections. Wired Devices have a fixed location, however, all devices must communicate via the network router.



(b) An example MANET topology for a set of laptops. Connections between laptops are indicated by bolt symbols. There is no dedicated router for the network, all devices can act as routers.

Figure 2.1: Example networks.

location, and the average home environment.

Wireless networking does not require any fixed connection. It allows for mobility within and across networks, because devices are not restricted by the physical length of cables found in wired networking. A mobile device may run on alternative sources of energy, e.g., solar or battery power. This enables networking in areas where networks previously could not sustain them: situations in which in areas without electricity or where nodes frequently change location. Examples include the developing world where infrastructure is unreliable or not present, and military scenarios where nodes frequently change location. Mobile networks can also enable new health-care scenarios, where patients can carry the device on their person, without obtrusive wires. There are scenarios where we do not have, or need, any infrastructure network: where nodes are mobile. The first area we discuss is therefore Mobile Ad Hoc Networking (MANET).

2.1 MANETS

Routing is important in any network because it enables messages to be sent from source to destination through intermediate nodes; routing provides a path for messages through nodes within the network. There are two routing strategies: *proactive* and *reactive*. Proactive routing is where the routing tables are constructed periodically, therefore, the route is ready when needed. This does not cope well, however, with topology changes due to mobility. In reactive routing, the routes are constructed on demand. Reactive routing is more useful in situations with mobility as effort is not spent on constructing routes that are later made inaccurate by topology changes (it is still possible, however, for the topology to change mid-flow, which can break the route).

MANETs are self-configuring networks, in which nodes connect via wireless links. Nodes are free to move around, therefore enabling the topology to change arbitrarily. We see an example MANET topology in Figure 2.1b. Rather than all nodes connecting to access points to route packets, all nodes in a MANET may be asked to route packets in a MANET.

There are two main MANET routing algorithms: Dynamic Source Routing (DSR) [76](Proactive) and Ad Hoc on Demand Distance Vector Routing (AODV) [117](Reactive). When using DSR the source sends out a single message to the nodes in range via broadcast; the destination address and an ID number are contained in the message. If they can provide a route to the destination the intermediate nodes will send a route-reply back to the sender. If the intermediate node does not know the route it will flood the network with the original message and its IP address appended to the list of addresses in the packet header. When the destination receives the discovery packet, it will reply with a route-reply message to the node it received the enquiry message from. When the source receives a route-reply it adds it to its route cache.

When sending data using AODV Routing, a source node checks if it knows a route to that destination. If the source does not have a route to the destination it will broadcast a route-request message. When nodes in the network receive a route-request message they will update their information about the source by creating a record of which node sent

them the message, they will also forward this message, unless they already have a route, in which case they will send a route-reply back to the sender using the route it received the route-request from. The sender will use the route that it finds to have lowest cost (usually measured in hops).

There are several other MANET protocols [153]:

- Optimized Link State Routing (OLSR) [29]— a proactive link state protocol. Nodes flood a topology table to all of the nodes within the network. Nodes then work out the optimal forwarding paths locally.
- Location-Aided Routing (LAR) [83]— these protocols use location information (e.g. GPS) to improve routing performance.
- EASE [62]— using encounter histories to improve routing.
- On Demand Multicast Routing Protocol (ODMRP) [87]— a reactive multicast protocol that uses mobility prediction.
- Dynamic MANET On-demand (DYMO) [116]— can work as both a proactive and reactive protocol. DYMO is intended to be a successor to AODV.

Each of the MANET routing protocols mentioned above however, assume that there is an end-to-end path between the peers in the network. It is possible to imagine a case where there is no point in time where two nodes are ever part of the same network partition, yet we consider them to be members of the same network. Perhaps a node runs out of energy, or moves out of range. It may be that a node powers down because it is only needed for certain periods of the day, e.g., during office hours. This schedule breaks that nodes connectivity to the rest of the network for an extended period of time.

One problem with MANET routing protocols is that they assume that a network partition has occurred if the path between a message's source and its destination is lost. This creates a problem for networks where the nodes only have intermittent connectivity, e.g., wireless sensor networks [34] where the nodes only have periodic connectivity to conserve battery power. In MANETs, if a node is not detected in the network at all times by at least one node it is assumed to have left the network.

If there is a large delay, existing transport protocols such as TCP can break down, even if the end-to-end connection is still active. The large delay causes the congestion control to assume packet loss. Another problem for TCP is that the window size might take a very long time to enlarge due to the wait for ACK packets. A different strategy is therefore needed for networks with intermittent connectivity or when there is a large delay.

2.2 Delay-Tolerant Networks

As we have discussed, the architecture and protocols of the modern Internet may not operate well in environments where there is a long delay or a sparsely connected network. These conditions are often found in networks with mobility and limited battery power, as well as in extreme environments [21]. Often called challenged networks, a delay-tolerant network (or Intermittently Connected Network [153] [ICN]) is a network where some of the assumptions of the TCP/IP stack regarding the performance characteristics of network links do not hold.

> "Applications in ICNs must tolerate delays beyond conventional IP forwarding delays, and these networks are referred to as delay/disruption tolerant networks (DTNs)." [153]

Examples of the assumptions of conventional IP networks that do not hold in the DTN domain are [48]:

1. An end-to-end path exists between peers in the network.
2. The maximum round trip time between nodes in the network is not excessive.
3. The end-to-end packet drop is small.

We find these properties, for example, in communication with Mars [21]. Similar properties emerge in very high speed networks where waiting for control traffic reduces performance, and in situations with a large amount of mobility. There may be mobile nodes in the network that have intermittent connectivity, e.g., commuters in a city may go through

areas with very different connectivity. Such examples include walking on a pavement compared with travelling on an underground train network.

As a result the DTN routing and data-delivery architecture is unlike conventional TCP/IP networks. DTNs use a store-and-forward architecture, where nodes forward a group of messages together. In DTNs the data unit can be a message itself, a packet or a bundle of messages. The network is split into regions, where links with high delay mark region borders. The internal region networks can use different mechanisms from the transport-layer down in the network stack. The network uses the DTN bundle protocol to communicate between regions on the high delay links. The bundle protocol sits between the application and transport layer in the traditional internet protocol stack.

Cerf et al. have put forward an RFC (Request For Comments) for a delay-tolerant network architecture [25]. RFC 4838 covers the routing and transport layer approaches:

- Virtual Message Switching Using Store-and-Forward Operation: allows nodes to exchange messages in high delay environments.
- Nodes and Endpoints: nodes may be members of groups called "Endpoints". When the bundle reaches a minimum subset of end nodes it is considered delivered.
- Endpoint Identifiers (EIDs) and Registrations: endpoints are identified by Uniform Resource Identifiers. Each node needs one EID that uniquely identifies it.
- Late Binding: allows the final destination nodes to be determined on route rather than at the source.
- Fragmentation and Reassembly: DTN nodes may divide up application data into bundles. The final destinations are responsible for extracting the application data from incoming bundles and reconstructing it.
- Reliability and Custody Transfer: the bundle layer can provide acknowledgements and prioritisation of traffic.
- DTN Support for Proxies and Application Layer Gateways: the bundle layer provides a common method for connecting application layer gateways.

- Timestamps and Time Synchronization: timestamps are used to expire bundles.

- Congestion and Flow Control and Security: are left as research topics.

2.2.1 DTN routing

In DTNs the latency associated with reactive routing approaches is less of a concern than in MANETs, as the delay due to the lack of connectivity is usually orders of magnitude greater than the delay associated with constructing an end-to-end route. The large inter-encounter period affords routing protocols with time to make relatively more complex decisions in comparison to MANET protocols.

We can break DTN routing protocols down into two classes [153]. The first for deterministic time-evolving networks, and the second for stochastic time-evolving networks. We class deterministic networks as those for which the routing topology can be pre-determined. We class stochastic networks as in which nodes only have a probabilistic chance of appearing within the network at a given point in time.

Satellite networks are an example of a deterministic DTN, because the position of satellites can be predicted in advance. The movement of ambulance crews provides an example of a stochastic DTN, because the ambulance crews must respond to external input (call-outs) and therefore they do not have a predictable schedule.

The most simple stochastic DTN routing strategy is "store-and-wait". The sender holds onto the message until it encounters the destination node directly. This is a cheap approach in terms of energy, as it preserves battery life, but the time to message delivery can be large or infinite in the case where two nodes never appear in the same network partition.

The antithesis of this approach is Epidemic routing [142], where an intermediate node, on receipt of a message, forwards it to all of its neighbours bar the sender. This implicitly assumes that the intermediate nodes will come into contact with other nodes closer to the destination through mobility. When this occurs the nodes swap messages that the other party has not seen before, hopefully leading the messages towards their final destination.

Many routing protocols use other nodes to forward messages. We refer to nodes physically carrying data as "data muling" [61]. Data muling can increase bandwidth, as Grossglauser et al. state:

> "Our results show that direct communication between sources and destinations alone cannot achieve high throughput, because they are too far apart most of the time. We propose to spread the traffic to intermediate relay nodes to exploit the multi-user diversity benefits of having additional "routes" between a source and a destination." [61]

As a result, DTN routing protocols are a variation on one of two approaches:

- Flood the network, sending lots of copies of the message, in the hope that some of the copies arrive
- Send a single copy (or a low number of copies) of the message, hoping that the information used to make the routing decisions is accurate

We have seen that DTNs solve the problem of disconnection and delay that MANETs are unable to cope with, by providing a store-and-forward mechanism that sits on top of the network transport layer. DTNs are not, however, appropriate for all disconnected scenarios.

2.3 Opportunistic Networks

DTNs are networks designed for systems with a few high delay links. There are also scenarios, however, in which all the nodes are subject to high-delay/disconnection. One example is humans commuting to work in a city. Almost all the nodes in such a scenario move between two distant physical locations: home and work. Even nodes that are seemingly in one physical location, e.g., a market-stall trader on the commuter route to work, have very few static connections. Most of the trader's connections are with commuters who occupy one location for a short period of time only.

This fully mobile stochastic scenario is different to a DTN scenario. Rather than regions with short intra-region delays and large inter-region delays, as in a DTN, there is the potential for high delay or disconnection on all links. An opportunistic network [115] is a store-and-forward network that can route messages in a high-disconnection environment. DTNs can be seen as a subset of Opportunistic Networks (as DTNs feature only a few high delay links). Hui et al. have created a "Pocket Switched Network" (PSN or Opportunistic network) [69], and describe networking in a scenario where humans carry small personal computing devices capable of exchanging messages. We can leverage the network from the encounters between the devices as the humans go about their daily lives.

Opportunistic networks can enable many new applications. Lindgren and Hui highlight several areas in which they think that opportunistic networking can make a positive improvement upon existing technologies [94]. They believe the developing world would be an ideal place for low-cost messaging and network access from mobile devices, where cellphone markets are increasing in size.

Urban areas in developed regions can also benefit from opportunistic networks. Hui et al. argue that even in the presence of existing infrastructure coverage, e.g., 3G networks, opportunistic networks could be of benefit when distributing large data items. In areas with low 3G coverage, opportunistic networks allow improved communication applications using hybrid opportunistic networks [73].

As well improving existing mobile applications, opportunistic networks open up possibilities for new applications that were not previously carried out on consumer electronic devices:

- Crowdsourcing and messaging [23, 105]
- Sensing [33, 44] and collaborative sensing [56, 136]
- Military surveillance [148] and human tracking [6]
- Sport-activity monitoring [43]
- Animal tracking [77]
- Personal sensing for healthcare monitoring [88]

- Mobile file sharing [102, 120]
- Interaction with embedded AI in pervasive environments [63]
- Opportunistic computation [30, 109]

Routing in an opportunistic network operates on the principle that, in the absence of connectivity to existing infrastructure networks, passing messages between devices carried by humans could eventually route a message to the intended destination. This relates to Stanley Milgram's famous small-world experiment [138] where he found he could route letters between two unfamiliar individuals over a large distance, one in Nebraska, the other in Massachusetts, in under six hops. Opportunistic networks have to been shown to have similarly small contact patterns [27].

The Haggle architecture [124] is a data-centric architecture for opportunistic networks. Haggle uses asynchronous connectivity, allows intermediate nodes access to ad-hoc multicast content, and exploits any available data transfer methods to take advantage of brief encounters using store-and-forward message switching. Hui et al. have highlighted that finding the correct groups of nodes to forward messages to aids in routing and node efficiency [70, 71] (we look at this problem further in Chapter 5).

The problems facing opportunistic networks include energy efficiency, user privacy concerns, incentivising participation, and routing. Energy efficiency is an important concern for protocol designers, as forwarding messages has an associated cost in terms of energy used. User privacy concerns may include the over-exposure of encounter data. Incentivising participation concerns the problem of ensuring that users participate in message forwarding. Routing concerns the selection of appropriate encounters to use for forwarding.

There are two main approaches for message forwarding in opportunistic routing. Using the first approach, we can send many copies and constrain the flood according to some mechanism, e.g., forward to all nodes who have recently encountered the destination. Using the second approach, we can limit the number of message copies (as few as one), but expend more effort computing which encounters to use for exchanging the message, e.g., use a detailed model of movement patterns to predict which node frequents the current

location and encounters the destination often. The data used for both approaches' computations routinely comes from storing a history of encounters. These stored encounters are then used to drive the routing algorithm.

Researchers look at collecting traces of human encounters to then use for routing simulation [24,39]. This is because it is costly to run experiments with large numbers of participants carrying mobile devices. By collecting and sharing encounter data, researchers can repeat one another's experiments [84,149]. Traces of human encounters can be more easily collected today than in the past, because most people in the developed world already carry mobile phones (81% of people in the UK [50]).

Opportunistic networks involve humans operating according to their daily schedules. The task of the routing protocol is to decide if the encounter taking place is appropriate for routing any of the outgoing messages. We can therefore see that studying the social connections between individuals can give us insights into the usefulness of encounters for forwarding.

When considering the incentives for nodes to participate in forwarding, we should remember that mobile nodes carry the devices in an opportunistic network. We can, therefore, assume that the devices run on battery power. The cost of forwarding is measured by the battery power the device uses to forward messages. An economically rational user will conclude that it is in their best interest not to forward messages on behalf of any other nodes (as this saves their battery power), while expecting other nodes to forward their messages. If this is the case, the network will not be able to forward messages; only messages exchanged directly with the destination will arrive. We therefore need a way to incentivise the users to participate in the routing protocol. This is an example of what economists refer to as the free rider problem [154].

2.4 Social Networks

Both opportunistic encounter patterns and traditional social networks represent a relationship between individuals. In an opportunistic network the relationship is an "encounter"

that takes place because individuals were co-located, in a social network the relationship is dependent on some kind of social connection. We shall see that opportunistic-network encounter patterns are also similar to social networks, due to similar graph structure and contact distribution.

To understand the usefulness of the similarity between social and opportunistic networks we first need to understand social networks. First we will discuss terminology; second, we will discuss how social networks form; third, we will discuss how social networks can improve opportunistic networks. Finally we see how to compare social networks.

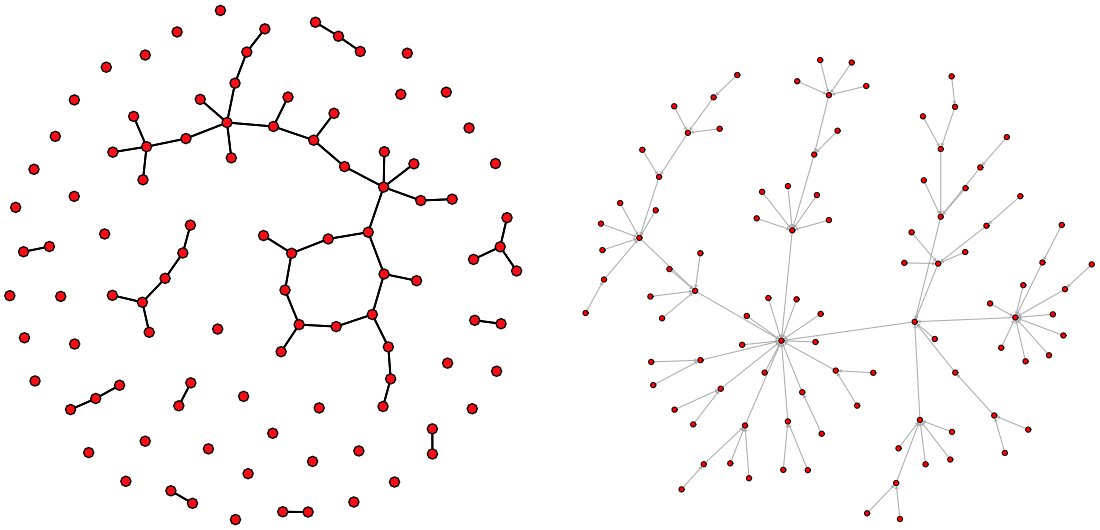
A social network is a social structure of individuals or organisations (vertices/actors/nodes). These entities are joined together via social connections of some kind (edges/ties); friendship or business ties are good examples. It is the patterns of relationships which are interesting, rather than the characteristics of the individual nodes [52]. The social network is routinely represented by a graph of the edges and vertices.

In this dissertation, we use the vocabulary from [52], and expand with relevant entries in the appendix A.

2.4.1 Social Network Models

What we consider social network analysis has arisen from the study of graph theory, and the study of community relations between humans [82]. To understand the structure of these relationship graphs, we need to understand how social networks form. We can consider a graph of a network as nodes, and ties between nodes. To generate a network we can assume that connections between nodes are made at random.

The most famous model of random graphs is the Erdős and Rényi model [45]. In this model, nodes have an equal chance of connecting to one another with a given probability. We can see an Erdős-Rényi model graph in Figure 2.2a. Erdős and Rényi's model however, does not capture the behaviour of social networks, as the distribution of connections does not match the expected power-law form.



(a) A graph created using the Erdős-Rényi random graph model. Here we see a links from a random uniform distribution.

(b) A graph created using the scale-free network model. The degree distribution follows a power-law.

Figure 2.2: Randomly generated graphs. Notice the presence of hub nodes in Figure 2.2b.

More recently, the scale-free network model [9] by Barabási and Albert, is believed to be more closely related to that of human social networks. Scale-free networks model the world-wide web [3, 68, 85], internet links [49], e-mail networks [42] and the power-law connectivity patterns also appear in opportunistic contact traces [26]. Scale-free networks also capture human sexual contacts [92].

Scale-free networks, rather than assuming the connections amongst a static set of N nodes, iteratively add nodes into the network graph. The principles are that the network is constantly adding new members, and nodes have preferential attachment (to nodes with a high degree), which creates nodes with a relatively large number of connections (hubs) as in Figure 2.2b.

Kwak et al. show that connections in online social networks also follow a power-law distribution [86]. The phone call network is similar to human encounters, as calls also follow a heavy-tail distribution [24].

Amaral et al. find that social networks of actor collaboration, Mormon acquaintanceships and friendships amongst school children, follow a heavy-tail distribution that al-

though similar to a power-law distribution, is instead a truncated exponential distribution [5]. They conclude that it is a combination of ageing of the vertices and limited vertex capacity (the physical cognitive limit on vertex numbers) that prevents the scale-free model from exactly fitting the social networks studied. Researchers have shown that humans have a cognitive limit on the number of individuals they can maintain knowledge the relationships between [37,133], placing the number at 147.8 [38]. A similar limit manifests in online social networks [2].

We can see that there are similarities between social, encounter, phone and online social networks. Can these similarities can be exploited?

A useful property of scale-free networks is that they are tolerant to random link failures, where up to 5% of nodes can fail before the network communication capability decreases [4]. Albert et al. find however, that scale-free networks are susceptible to targeted attacks on the hubs, where a loss of 5% of nodes results in network diameter doubling.

2.4.2 Social Network Analysis

Current research trends include community or graph properties. If we understand the structure of an encounter network, we can potentially gain insight into the performance of an opportunistic network using the encounter network.

Newman looks at finding community structure in networks [110–112], and proposes algorithms to find groups and discover community structure. A community is as a set of nodes which has many ties within the group, but very few ties to nodes in other groups. Communities are useful because we can assume similar behaviour amongst members of a community [54]. Ties that bridge communities are important as we can exploit these ties for message routing [36,59,75].

As well as evaluating encounters, any opportunistic routing protocol should consider which nodes to select for forwarding messages. The choice often depends on the result of a metric for comparing nodes in a network. If we can compare nodes using a metric, we can rank nodes. We can use the ranking of nodes to decide which node is more useful

for forwarding. Daly et al. discuss several metrics [36] naming closeness centrality and betweenness [53] to be some of the most important performance metrics for opportunistic network routing. As well as closeness centrality, clustering coefficient is important since it is another measure of the similarity of nodes. A higher clustering coefficient means that the network shows greater "cliquishness". We can think of clusters as communities within the network.

> "A network is said to show clustering if the probability of two nodes being connected by a link is higher when the nodes in question have a common neighbour" [36].

If we can work out which are the "important" nodes within the network (hubs and nodes with weak ties), this information can aid the design of routing algorithms to forward messages closer towards the destination [46, 70]. We can also use this knowledge to prevent the overloading of nodes. If we maximise the lifetime of nodes, we avoid dividing the network [28].

Collecting encounter data can be costly for the researcher. If we can show that social networks and encounter networks are similar, then insights we make into social networks could be applied to encounter networks. We could, therefore, use this information to improve opportunistic networks, specifically by enabling protocols to make informed routing decisions without having to collect large encounter histories.

There is further evidence to link the models of opportunistic networks and human social networks. Srinivasan et al. present a study on the contact patterns of students [130], analysing their timetables. They study the inter-contact times and the spread of computer viruses in such networks, which is similar to Su et al. [134]. Srinivasan et al. find that:

> "The student population is well mixed, with a small degree of separation between them. This observation is good news for mobile applications that rely on proximity-based connectivity to disseminate messages, as a potentially large number of contacts can be made." [130]

By studying the epidemic spread of information inside this social network of students,

Srinivasan et al. find that: "arbitrary pairs of students can communicate with each other in less than two business days on average." [130]. This indicates that many opportunistic networking applications with a TTL of less than two days are likely to perform worse than those with a TTL higher than two days.

Khelil et al. describe some contact-based metrics for the analysis of social networks [81]. Specifically, they mention contact rate, encounter frequency, encounter rate, encounter duration, contact loss duration. With these metrics we can understand mobility on a large time-scale.

Rather than using encounters we have collected between individuals, we can also consider networks where the nodes provide a "list" of nodes that they consider themselves to have a tie with. These can be considered ***self-reported social networks (SRSN)*** following the terminology of [41]. We can consider networks where we infer the connections between nodes from contacts or communication, as ***detected social networks (DSNs)*** following the terminology of [74, 110].

It is well known amongst social science researchers [10, 32], that self-reported and detected social networks differ. A recent study by Mtibaa et al. [106], however, looked at self-reported and detected social networks amongst a group of conference attendees, finding that the two social networks are similar. These results however, may be peculiar to conference environments. Regardless of the similarity or difference between the two types of social network it is still interesting to see how useful SRSNs can be for improving opportunistic networks.

If social networks and encounter networks are similar, then inferences made about one network may be relevant to the other. We may therefore, be able to sample social-network information instead of encounter information, and come to the same conclusions without performing a field-experiment, thus saving time and money. If social and encounter networks are different, inferences about one may improve our understanding of the other. For example, we might observe that all members that bridge groups in an encounter network are from one group in the social network. If this were true, we could then design a routing protocol to make particular use of these group of nodes.

2.4.3 Comparing Nodes and Networks

If social networks are similar to contact networks, the ability to compare networks is important for developing and understanding expected node behaviour amongst the nodes in the network. For example, if we consider two nodes to be similar, we can save routing information by only storing one set of behaviour predictions for both nodes.

Routing protocols select nodes to use for forwarding messages. Understanding similarities between nodes in the network may be useful for routing. We therefore now look at computing the similarity of nodes. First, we consider how similar the nodes are to one another in terms of which other nodes that they are in contact with. Second, we consider how to calculate how similar the nodes' positions in the network are in terms of how they are connected to the other nodes' positions in the network.

Structural Equivalence

We can view nodes as occupying a position in the network, given by the set of ties they have to all other nodes in the network. We can then see, that it is possible for two nodes to have a measure of similarity of ties. If we compare the relations of nodes we obtain a measure of the structural similarity of the nodes. Euclidean distance is a common metric used to measure structural similarity [22].

It is also possible to use structural equivalence to compare node properties. Blockmodelling for structural equivalence [145] groups nodes together into structurally equivalent sets called blocks. We can compare the sets of blocks to see the differences between multiple dimensions of the same data, e.g., call records and Bluetooth records for a mobile phone trace. We can thus make inferences about the difference and similarity of nodes.

As there can be many possible blockmodels with even only a few types of tie, Sailer developed a hierarchical method called CONCOR [123]. CONCOR requires the researcher to decide the number of hierarchical levels required. We cannot therefore, easily automate a system to use CONCOR. Sailer describes the concept of structural relation, which he sees not as a tie between specific people, but as a set of ties in an entire population. In his view,

the network is the patternising of the social relations over a set of persons positions or groups. This approach is similar to regular or role equivalence.

Regular Equivalence

Structural equivalence tells us how similar two nodes are to each other, but, it does not consider the types of relationships between nodes. We can use regular equivalence to do this.

Borgatti and Everett [17] describe Regular Equivalence as a partition of nodes into classes such that nodes of the same class are surrounded by the same classes of nodes. We can consider the classes as "roles" in the social network. They have developed an algorithm called CARTREGE which takes a multiplex matrix (see [144, p 208]) and works out the regular equivalence in a neighbourhood of nodes. At first, CATREGE assumes nodes are equivalent, then it iteratively breaks the nodes down into separate role classes.

We can view the roles as groups of nodes that are similar to each other in terms of their relationships to other nodes. For example, nodes in the same role may all bridge different groups of nodes in the social network.

Exact role assignments, however, are a case of regular equivalence where the nodes have to have the same number of occurrences of another role in their neighbourhood to form that role. These are largely difficult to find in practise.

We therefore need a way to deterministically work out the role assignments in the network, without any manual intervention.

2.5 Incentives For Participation In Opportunistic Networks

Opportunistic networking relies on cooperation between nodes, that is, the users participating in the network, to perform efficiently. Opportunistic routing protocols depend on nodes forwarding messages for each other, as otherwise the only delivery mechanism

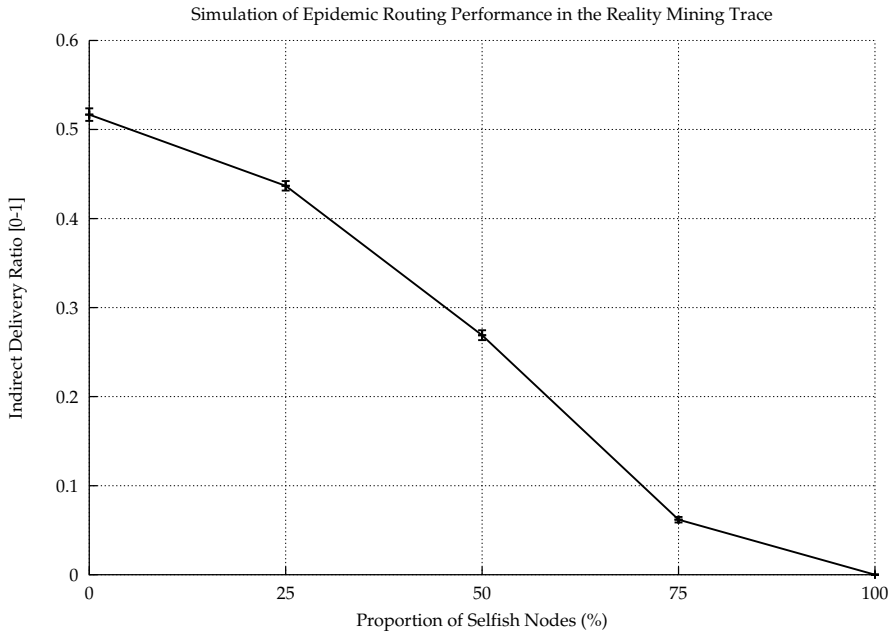


Figure 2.3: Simulation of an opportunistic network application using Epidemic routing [142], and the Reality Mining mobile-phone trace [40]. Indirect delivery ratio is defined as the number of messages delivered that were not passed directly from the source to the destination, divided by the number of unique messages. As the proportion of selfish nodes in the network increases, network performance in terms of delivery ratio decreases. Selfish nodes do not pass messages that originate from nodes other than themselves. Error bars are included, but are too small to be seen.

would be for the creator of a message to encounter the message destination node and deliver the message directly. Cooperative forwarding, however, incurs a cost to the forwarding nodes, both in terms of energy (battery power) and storage (the space required to store forwarded messages). Both of these are a constrained resource in mobile devices such as those used in opportunistic networks.

Due to these costs, nodes may wish to avoid the costs associated with participation in an opportunistic network, by not forwarding messages for other nodes. We call this behaviour selfishness. Nodes behaving selfishly attempt to avoid costs to themselves, at the expense of the performance the other nodes receive from the network. When a node is behaving selfishly, it will not forward a message for another node at any time.

Figure 2.3 shows the results of an opportunistic network simulation where nodes act selfishly. In this trace-driven simulation, using the Reality Mining opportunistic data

trace [40], nodes create and forward messages according to the Epidemic routing protocol. We performed runs with differing numbers of selfish nodes. We see, that as the percentage of selfish nodes is increased, the performance of the network, i.e., the number of messages delivered, decreases rapidly as the percentage of selfish nodes increases. If we can detect and discourage selfish behaviour, it might be possible to achieve the same performance as if no nodes are selfish, even if the all the nodes have a propensity for selfish behaviour.

How can we create incentives for nodes to cooperate? Incentives, reputation and trust have been extensively studied in peer-to-peer networks and MANETs, and more recently, in sensor networks and DTNs.

Huang et al. discuss the drawbacks of mobile DTN systems and state "without sufficient nodes cooperating to provide relaying functions a MANET cannot function properly" [67]. They go on to discuss the different drawbacks with various types of incentive systems for user co-operation. They find that in the early stage of a technology's development it is not necessary for the system to incentivise co-operation. They do however, state that this may be a problem in the future if the technology is to be used in a mainstream market.

We can view incentivising participation in the scope of malicious/selfish behaviour, as attacks and a lack of incentive for participation can both lead to reduced network performance. We now discuss attacks on opportunistic networks, as we cannot consider the improvements these networks can provide, without considering whether opportunistic networks are capable of providing those improvements in the face of a malicious attacker's behaviour.

To incentivise nodes to participate in forwarding, we can create an undesirable consequence for not following the routing protocol, or we can encourage participatory nodes through rewards. Opportunistic networks are mobile and ad hoc. The incentive mechanism thus needs to operate in a distributed fashion. For nodes to know whether or not a particular node is malicious or altruistic, we need to provide a common scale that nodes can be compared against. To do this we can incorporate the concept of a computable reputation score for nodes, thus enabling the nodes to punish or reward nodes for their behaviour.

2.5.1 Selfishness

If users are selfish, it harms the network as the nodes' messages may not get through. If all nodes were selfish, the only way to deliver messages would be via direct delivery. The users' economically rational desire to preserve their battery power affects this selfishness.

Many different mechanisms exist to create incentives to discourage selfishness [97]: from bartering (a direct exchange of services), to currency (behaviour that benefits other users earns measurable credits exchangeable for services). In the middle of this spectrum lies asynchronous bilateral trading: nodes perform actions to benefit one another, but not necessarily at the same point in time.

In a credit based system nodes must perform actions (altruistic behaviour) that they use to accrue credits, which they exchange for ability to forward messages [155]. If nodes can purchase more credits for the system they can afford to be more selfish: behaviour that does not benefit the network. If nodes all purchase more credits instead of following the routing protocol, then we are left with the same results as if there were no incentive mechanism at all. The cost of credits should therefore be set appropriately high.

To combat selfishness, many mechanisms incentivise participation by encouraging the other nodes not to forward the selfish node's messages [7,20,64,97,98,100,103,125]. Either selfish nodes do not have access to credits, or other nodes do not forward their messages because their reputation is below a certain threshold.

If the incentive mechanism incentivises nodes to behave in a network it is possible that the nodes may start to behave selfishly again if they can avoid detection for a period of time. It may also be possible in mechanisms without credits, that nodes only stop being selfish while they are detected as being selfish, and then resort to being selfish again once they are no longer considered selfish. There may be a window in which nodes that are being selfish are not detected as such. We need to look at how well mechanisms detect the selfishness, as this detection-delay exploit may be quite large.

Incentive mechanisms in which nodes gain reputation/trust by being on the delivery path, may encourage nodes to drop older messages. Nodes would be incentivised to drop

messages that offer them a minimal chance of reputation improvement. This reduces network performance because less messages get through to the destination.

It may also be the case that nodes are selfish to the point where they do not forward messages on behalf of other nodes, but will pass messages to the destination if they meet them.

2.5.2 Exploiting the friendship mechanism

Incentive mechanisms that use the concept of friendship or node pairing must take care not to incentivise nodes to accept all other nodes as friends or accept no nodes as friends. This can occur if the incentive mechanism hands out extra credits to nodes that are "friends" with the individual paying the credit charge, or perhaps by increasing the payout to individuals who are not friends with the node charged for sending the message.

2.5.3 Epidemic behaviour to gain trust

It may be possible for a node to improve its score by ignoring the routing protocol and flooding messages on all network paths. This would increase the offending node's standing in other nodes' eyes as they are being helpful. If all nodes took this approach, however, the network would experience reduced delivery ratios due to flooding the network . This attack is still economically rational, as a node could decide to be selfish when it needs to be (when experiencing low energy), yet can gain significant trust/reputation by forwarding when it has a large amount of energy, e.g., when charging.

2.5.4 Control traffic manipulation

In this attack, nodes may refuse to submit their credits or not exchange control information [18,55,96,155]. This behaviour makes it harder for the reputation mechanism to detect the selfishness, as there is less information to rely upon. Nodes may also try to offer routes that do not exist, by advertising that they are on the optimum forwarding path [18,20,152].

Many incentive mechanisms assign a high reputation value to a node that is doing a large share of forwarding. By reporting to be doing a higher share of forwarding than they are, the attacker can increase its reputation, as well as potentially avoiding its share of required message forwarding.

Nodes may create false acknowledgements of messages arriving at the destination [20]. This can be an attempt to avoid nodes asking the attacker to forward messages by causing other nodes to drop the messages that they may be holding.

Nodes may choose to create fake error messages, by claiming that a message was corrupt [18]. This would allow them to avoid having to forward any more copies of the message that they have been given.

2.5.5 Route manipulation

Nodes may try and alter the speed at which they perform periodic control processes, for example increasing the route update frequency. This may affect the weighting of reports for example, or cause increased energy consumption [18].

Nodes may alter the intended path of a message by forwarding it to a node that was not on the intended path [18]. This can be an attempt to decrease the chance of a message arriving at the destination. This is also a way for an attacker to cause a node to have to do more than its fair share of forwarding.

Nodes can try to claim that another node was or was not on the path of a message. Attackers attempt to create the appearance that a node has a higher (or lower) reputation that it does [98,155]. Nodes may also remove another node from the routing path to increase its own share of the credits or to discriminate against another node. By discriminating against a particular node, the attacker may hope to gain increased reputation, as this may increase its own likelihood of being asked to forward messages in the future, and therefore gain credits.

2.5.6 Defamation

Nodes may choose to create false reputation claims about nodes, e.g., claiming that a node is malicious when it is not [96,135,152]. This type of attack allows the attacker to potentially increase its own probability of nodes giving it a message in the future. This would allow the attacker to intercept more messages.

2.5.7 Tailgaiting

Nodes can follow another node, to create the appearance of a social connection or to collect records of encounters. In some systems nodes use encounter records as a form of credit [20, 90, 91]. If the attacker can increase the amount of time that it spends with a node, then the other nodes in the network may give the attacker messages for the intended victim of tailgating. The attacker may then refuse to deliver these messages to the victim node, effectively cutting it off from the network.

2.5.8 Collusion

Nodes may collude and work together to perform attacks [96, 152]. Rather than being an attack itself, this is a way for nodes to modify an attack strategy. Collusion adds another level of consideration for distributed reputation and incentive mechanisms designers to consider. For example, if nodes can pollute the reputation mechanism with false reputation reports, it may make accurately detecting malicious behaviour difficult.

2.5.9 Sybil attacks

Malicious attackers may create several fake identities to mask their behaviour or alter the outcome of any reputation mechanism or routing decisions [135, 152]. This allows the attacker to use a new identity in the network if the reputation mechanism detects their previous malicious behaviour. In a sybil attack, the attacker uses multiple identities to increase the reputation of the attacker's main identity, in the same way as multiple nodes

in collusion would.

2.5.10 Selective maliciousness

Nodes may try to beat any reputation mechanism by only being selectively malicious. Either in the time dimension (by altering at which times they are malicious) or the node dimension (by only being malicious to certain nodes) [135].

By performing attacks on only a subset of the nodes (or for a short period of time), the attacker may avoid detection, while still gaining some return from its malicious behaviour.

We can see a summary of these attacks and some defense mechanisms against them in Table 2.1. Not all of the attacks discussed are a relevant threat to all opportunistic networks, as not all opportunistic networks rely on the same routing or incentive mechanisms. Many of the attacks can be prevented with adequate security protocols, however, it is important that the network can provide the intended service to users. If users are not incentivised to follow the routing protocols, nodes will not follow the routing protocols, thus, research into prevention of attacks against the routing protocols would be irrelevant. We therefore see incentivising participation and avoiding selfishness as the most important of the challenges.

2.6 Summary

In this chapter we have seen how the need for opportunistic networking research has emerged from the study of challenged communication environments, and how opportunistic networks can be useful. We have noted the following points:

- We have motivated the need for opportunistic networks through example applications
- We have motivated the need for specific opportunistic networking research instead of using existing networking technologies

Table 2.1: Opportunistic Routing Threat Model

Attack	Defence
Selfishness	An Incentive mechanism
Traffic deviation	Watchdog nodes
Fake Routes	Can be allowed within reason
Fake error messages	Reputation mechanism
Increase Route update frequency	Encounter tickets
Silent Route Change	Not applicable as we do not have static routes
Collusion	Remove distributed mechanisms
Tailgaiting	Solve out of band
Adding/removing nodes on path	PKI
Submission refusal	Detected by reputation mechanism
Sybil attacks	SRSN can solve
Trying to live in the grace period	Better mechanisms to detect

- Opportunistic networks and social networks share similar properties, and understanding one can improve understanding of the other
- We have seen that the main challenge for opportunistic routing is how to decide which encounters to use for forwarding
- We see that of the several potential attacks against opportunistic networks, incentivising adherence to routing protocols is the most pressing

In the next section we see how current research addresses several of these points.

Chapter 3

Using Social Networks and Opportunistic Networks Together

In this chapter we assess the extent to which current research addresses several of the problems discussed in Chapter 2. First, we will discuss how social networks are used in opportunistic routing, and how we can improve routing using SRSNs. Second, we will discuss the problem with existing protocols being reliant on encounter histories. Finally, we will discuss how existing incentive mechanisms are inappropriate for opportunistic networks.

3.1 Social Networks For Opportunistic Routing

Opportunistic network routing relies on nodes forwarding messages to one another. The intention is that when the nodes forward the messages, the messages move closer toward the intended destination. For a high probability of delivery, we can exchange messages during all available encounters. We assume that when we flood the network, that messages eventually arrive at the destination (assuming that there are no constraints on buffers or energy), because all links will eventually be used. Epidemic routing [142] uses this approach. Messages are only forwarded to nodes that do not already have a copy of the message.

Intuitively we can see that a flooding approach is inefficient. Why should we use all links for forwarding? We should only forward messages along the shortest path (in hops and time) to the destination. Forwarding the messages directly to the destination is guaranteed to use the minimum number of hops (one). When compared to Epidemic routing, however, direct-delivery to the destination results in higher delivery times. We could therefore conclude that because Epidemic routing necessarily finds the one of the possible paths (but with considerable overhead), that it should be used for forwarding. But opportunistic network nodes do not run in constraint-free environments. They have a maximum buffer size, and a finite energy source. Epidemic routing has been shown, both intuitively and experimentally, to be energy expensive [127].

If Epidemic routing is too expensive, and direct delivery too slow, then we need to develop more efficient routing protocols that use the best quality links [46]. Identifying these links is an NP-hard problem [8].

To limit the number of links used for forwarding we can limit the number of message copies and links used in the network. A protocol that reduces the flooding nature of Epidemic routing in this way is Spray-and-Wait routing [128, 129]. The sender forwards a few copies of the message (around 12% of the number of nodes) into the network, and these intermediate nodes may then forward the message to the destination only. [1] This is a rather naive approach, as it does not use any potentially useful information regarding the encounter network itself.

Rather than place an arbitrary constraint on the number of message copies, the RAPID routing protocol [8] intentionally optimises a routing for a particular metric. For example, it is possible, to maximise the minimum delay, by viewing it as a resource allocation problem, rather than a link selection problem. This approach is appropriate for optimising routing with respect to one metric. We can imagine, however, that in an opportunistic network deployment, performance should be measured via a combination of several different metrics (including delay, probability of message arrival and effect of forwarding on the energy levels of nodes).

[1] There are extensions to this protocol that allow intermediaries to forward messages. One example extension allows nodes to forward based on a utility derived from recency of interaction [129].

It is because of efficiency concerns that researchers have designed protocols driven by the information nodes can collect about their environment. For example, we can use the frequency of nodes attending a location to drive routing decisions [58, 60, 102, 131]. While this is useful, more efficient protocols use information that the nodes collect about the encounter network itself to drive routing. If this information is collected by the device itself, then we see that nodes do not need a concept of global knowledge. Picu et al. find that when using local information, distributed protocols do converge [118].

Many existing protocols, such as Habit [101], use metrics based on locally-available information. By taking advantage of the encounter network structure and properties we can develop more efficient routing protocols. Lindgren et al. demonstrate their routing protocol PROPHET [93], and show that using encounter histories to predict future node encounters leads to equivalent delivery performance to Epidemic routing, with intermediaries required to do less forwarding. With MaxProp routing, Burgess et al. also argue for the use of a history of previous encounters to drive the routing decisions [19]. They show that another useful approach is to flood delivery acknowledgements to clear redundant message copies from intermediary nodes' buffers, because this leads to increased delivery percentages.

To improve routing performance further, many protocols use social network analysis techniques. Boldrini et al. find that users' sociability information (connections to users outside of their home group) can be used to improve opportunistic routing [16]. They also construct a middleware using contact-history based social information to improve opportunistic services [15], and integrate it into the established opportunistic architecture, Haggle [124].

Daly et al. present the SimbetTS routing protocol [35] that determines whether a node should forward a message to an encountered node using a combination of three social-network-analysis measures: *betweenness* [53] (roughly the number of shortest paths on which a node lies), *similarity* (the number of ties that two nodes have in common) and *tie-strength* (the recency, duration and number of interactions between two nodes). As nodes encounter each other, they share encounter histories, update their utility scores (the normalised sum of these three measures), and use these scores to determine whether to

forward a message to the encountered node. Tie strength is also used for routing by Fabbri and Verdone [47], and find that it offers a good trade off between performance and traffic volume.

The use of *centrality* for forwarding is also studied in [147], where it is used as a measure of node importance in the network. Xu et al. find, however, that when using this approach, a small number of nodes have to perform a large percentage of the forwarding (63% of traffic by 10% of nodes). Pujol et al. use interaction strength for routing [121], they find that in SimBet routing that around the top 10% of nodes carry out most of the forwarding (54% of all the forwards and 85% of all the handovers). We see then, that existing opportunistic-routing protocols using social-network analysis approaches are not efficient.

As we have seen, most protocols use information from the encounter network, (detected social network), but these are not the only types of social network. We could use pre-existing social network information (SRSNs). Many applications e.g., Facebook[2], Twitter[3], Last.fm[4] etc, ask users to explicitly declare their social network. We can consider how these social networks could be used for routing.

If we use an SRSN to obtain data for routing instead of DSN data, this can provide many benefits. If we validate protocols using the SRSN then we do not need to collect extensive encounter traces for experimentation. This also saves the user resources as their device is not expending energy on collecting encounter data. As SRSN data is indicative of the belief of a connection by the nodes (rather than based on a connection derived from proximity), SRSN based routing predictions may be more accurate than using the DSN data. The SRSN data reflect a longer-lasting concept of a relationship between individuals than the DSN, which is more appropriate in the short-term. This saves energy lost by redundant or unnecessary forwarding decisions made by the less accurate protocols. SRSNs provide better user privacy, because the users have explicitly provided (or consented to) access to the data used by the opportunistic routing protocol, rather than devices collecting and exposing data the user may not want to be used/exposed.

[2]www.facebook.com

[3]www.twitter.com

[4]www.last.fm

Mtibaa et al. [108] make use of SRSN information from online social networks to compute node rankings. Their PeopleRank protocol achieves an end-to-end delay and a success rate close to those given by flooding, while reducing the number of retransmissions by 50%. Similarly Hui et al.'s studies on forwarding and mobility [70,71], argue for using pre-existing social network information, and they create a routing protocol BubbleRap [72]. By ranking nodes within a community, and having hierarchical communities, BubbleRap allows nodes to "bubble up" a message to a node if it has a higher rank within the same community, or is a member of a community that is closer to the destination node. This external information provides useful information that aids forwarding, and might not be discernible from the encounter network itself.

Pietiläinen et al. describe an architecture that uses SRSNs to bootstrap an opportunistic network [119]. Nodes use their SRSN as a list of permitted intermediaries, and can decide to add nodes to this list over the network lifetime.

Mtibaa et al. find a strong correlation between the SRSN and DSN and describe a way to process the SRSN data so that they are more useful [107]. The SRSN and DSN data are mixed to improve routing.

Hossman et al. look at how well the social network analysis techniques (such as betweenness) are applied to the social graph [65], they find that for both SimBet and BubbleRap, the techniques are not applied to an optimal/efficient graph on condensed connections. They find that the graph could be improved by using a density based aggregation: for a given network density we can pick the best encounters to use. We thus see that we may be able to make better use of the encounter data for forwarding.

As well as aiding routing, social networks can be used for improving mobile applications. Miklas et al. [104] use social networks to improve three different mobile applications: opportunistic routing, mitigating the spread of worms, and mobile peer-to-peer file-sharing. Social networks and communities have also been used to build improved publish/subscribe architectures [66,150].

We have not seen an experiment which detects whether or not SRSN data is better or worse than DSN data for forwarding in an opportunistic network. We have seen that there

is no research into using exclusively SRSN data for forwarding. This may be because the focus before has been primarily concerned with proving that social networks can be used to improve routing. For this reason many of the social networks used in these routing systems were generated from encounter traces. A gap therefore exists in investigating how exclusively using an SRSN declared by the individuals taking part in the experiment can improve routing.

3.2 Encounter Histories And Over-Reliance

The nodes in an opportunistic routing protocol do not run in a constraint-free environment. Due to the finite buffer and energy stores of the nodes, routing large amounts of traffic though a subset of nodes can lead to nodes batteries becoming drained.

We have seen in Chapter 2.4.1 that if a certain proportion of the hub nodes are removed from the network, the network diameter can increase, leading to longer forwarding paths and in turn, more nodes running out of battery. As discussed in the previous section, we have seen that existing opportunistic routing protocols require a small number of nodes to do a large proportion of forwarding [121, 147]. There are, however, no protocols that sufficiently address this problem.

As the nodes that perform forwarding are the hub nodes [121], and due to the clustered and assortative nature of opportunistic networks [137], routing protocols that rely on these nodes will cause nodes to run out of battery, thereby reducing performance. We therefore need an opportunistic routing protocol that does not rely on a subset of nodes to perform the forwarding. Perhaps we can use social network analysis techniques to find nodes to use as alternatives for those nodes that are required to do a lot of forwarding.

As we have seen in Section 3.1, many existing routing protocols rely on an encounter history for information used in routing decisions. Using an encounter history to make an informed decision, however, means that a number of encounters must take place before the routing protocol works as intended. Thus, the SimBetTS authors propose that a period of time (15% in their simulations) be reserved as a "warm-up" period, during which nodes

gather encounter information, but do not forward messages [35]. The PROPHET protocol similarly uses a warmup period.

We see that there is a need to develop routing protocols that can operate from the time of network startup, without relying on the encounter histories for forwarding.

3.3 Incentives For Participation

We have seen in Chapter 2 that there are many attacks against opportunistic routing. Burgess et al. however [20], show that opportunistic networks are robust to many attacks, without even having to use authentication. They find that because the routing protocols frequently use a lot of message replication, opportunistic networks perform well in the face of attacks.

We also need to consider the incentives that users have to participate in a routing protocol at all. It would be rational for nodes to forward their own messages at all possible opportunities and to expect other nodes to forward all of their messages, whilst they themselves do not forward messages on behalf of any other nodes. While rational, this behaviour is selfish, to avoid costs, or to reduce congestion due to other nodes traffic, so that the selfish-node's messages have an increased likelihood of reaching the destination.

The vulnerability of opportunistic networks to selfishness is discussed in [79]. Karaliopoulos finds that DTNs are vulnerable to selfishness, and produces a metric for assessing the vulnerability, called the "deceleration factor". Panagakis et al. [114] also show that the opportunistic routing protocols can be susceptible to selfishness, in particular Epidemic and Spray-and-Wait.

Keränen et al. find that once around 30% of the nodes in the network are selfish, the performance of the routing protocols degrade [80]. Shevade et al. find that the performance of the network can drop to as low as 20% if nodes are selfish [125]. Thus, rather than looking at attacks on the network, we focus on the incentives for participation in opportunistic routing.

Xu et al. [146] look at the different distributions of altruism for opportunistic sharing. They find that nodes with lots of contacts cannot maintain a uniform altruism to all connections due to resource constraints. They find that the "important" nodes should be favoured, as they are responsible for maintaining the short diameter of the network. This fits with our view of certain nodes in the network (hubs) having more influence over the transmission paths of messages than the average node. Solis et al. describe a mechanism that uses classification for the nodes to prioritise the traffic of nodes that contribute towards routing [126].

Sun et al. develop a trust measure for distributed networks [135], and argue that a trust measure is necessary to facilitate cooperation.

When Resta et al. look at node cooperation [122], they find that even a small number of nodes co-operating leads to significant improvements over the case where all nodes are selfish. If we can incentivise a small proportion of nodes to participate it could lead to increased routing performance. We need to decide however, how to incentivise this participation.

To combat selfishness, we must first detect it. Several approaches use "watchdog" mechanisms [7,18,64,99,100,103], where a third node oversees a message exchange between two nodes to verify its authenticity. Such an approach, however, is inappropriate for opportunistic networks, as routes are rarely static and the inter-contact times are large; neighbours are not consistently available to monitor the behaviour of one another.

For a disconnected opportunistic network, it is therefore necessary to rely on the encounters between nodes as the only way to exchange data and incentive-mechanism control traffic. The most common detection approach for opportunistic networks is for all nodes to monitor their own encounters, and to exchange their opinions of other nodes when they interact. Nodes then use these collated opinion data to make decisions about the trustworthiness of individual nodes. Liu and Issarny [96] describe a recommendation model, which is used to allow nodes to share their opinion of nodes so that malicious behaviour can be detected.

Liu et al. argue in favour of using reputation of recommendations as well as reputation

of service provided by nodes [96]. They discuss how to force nodes to reveal the control information and reputation information.

Yu et al.'s reputation system for peer-to-peer (P2P) networks has nodes build opinions of other nodes by analysing the quality of service (QoS) that they receive from these nodes [152]. They also use a measure of credibility for the nodes in the network.

They include a rating-discovery algorithm that maintains consistency of ratings across the network. P2P networks, however, have different properties to opportunistic networks. Even though there is potentially high churn in peer-to-peer networks, it is generally assumed that direct connectivity between any two nodes in the peer-to-peer network is possible, which is unlikely to be true in an opportunistic network. In opportunistic networks, it is harder for nodes to establish the opinion of nodes, as they cannot connect to arbitrary nodes to request reputation information.

To verify the reputation information, we must be able to prove that the reputation is based on experiences – how can we prove that messages were exchanged, or that encounters occurred? One way to validate encounters is to use encounter tickets [89], a cryptographic mechanism that nodes can use to prove encounters and message exchanges took place. When nodes two nodes meet they create a data item containing the timestamp of the encounter, known as an encounter ticket. Assuming the nodes are using PKI, both nodes sign the ticket with their private keys. This provides non-repudiation for encounters. Encounter tickets allow nodes to build up a history of message exchanges to use to construct an opinion of other nodes. Nodes can exchange encounter tickets and opinions during encounters.

SMART is a credit-based multilayer incentive mechanism for DTN [155]. SMART uses provable exchanges and layered coins. This protocol requires the nodes to connect to a trusted authority for credit checks for nodes to send messages. Lu et al. also use an encounter-ticket-based incentive mechanism [98], but this requires a trusted authority (an out-of-band oracle), which makes it inappropriate for opportunistic networks. Li et al. propose RADON [90], which uses a history-based approach, combined with watchdog nodes. RADON floods control messages to the network, potentially consuming lots of nodes' resources.

Any incentive protocol that relies on trust and reputation needs initial reputation values for the nodes [125]. As SRSNs provide information that is available before network startup, then perhaps we can use SRSN information to bootstrap the incentive mechanism. Social community information has been used to thwart node capture attacks [31].

The RELICS protocol [139] encourages cooperation through ranking. Nodes estimate the likelihood of message delivery for each of the nodes they encounter, and use this to rank nodes. A node's rank is improved by being on the forwarding path of successful delivery. Nodes adjust their energy expenditure to meet a desired delivery ratio threshold (decided *a priori*). By expending more energy (forwarding more messages), nodes can hope to deliver more messages, increasing their rank with other nodes. Similarly, if their delivery ratio is above the threshold, nodes drop their energy expenditure. The RELICS protocol requires nodes to flood control messages to the network. It may be the case that not all of the nodes receive the control messages, and potentially there is a large energy overhead for control messages.

3.4 Summary

This chapter has outlined the current research in the area of opportunistic routing and incentivising participation in opportunistic routing. We have noted the following points:

1. Social network theory has been shown to be useful for opportunistic routing
2. Existing protocols using social network information result in certain nodes becoming overloaded
3. SRSNs might provide information useful in efficient routing protocols
4. There is the problem of bootstrapping opportunistic routing protocols
5. While work in incentive mechanisms exists, current opportunistic network incentive mechanisms require infrastructure connectivity or oracles to operate

We believe that by using SRSNs, we will be able to address these problems. In the next chapter we perform an experiment to demonstrate that SRSNs can be more useful than

detected social networks for opportunistic routing.

Chapter 4

Self-Reported Social Networks for Opportunistic Routing

Opportunistic networks may form an important part of future mobile-computing environments. Understanding how to efficiently and effectively route information through such networks is an important research challenge, and much recent work has looked at detecting communities and cliques to determine forwarding paths.

Such detected communities, however, may miss important aspects. For instance, a user may have strong social ties to another user that they seldom encounter; a detected social network may omit this tie and so produce sub-optimal forwarding paths. Moreover, the delay in detecting communities may slow the bootstrapping of a new delay-tolerant network.

This chapter explores the use of *self-reported* social networks for routing in mobile networks in comparison with *detected* social networks discovered through encounters. Using encounter records from a group of participants carrying sensor motes, we generate detected social networks from these records. We use these networks for routing, and compare these to the social networks which the users have self-reported on a popular social networking website.

A fundamental problem for opportunistic networks is how to effectively and efficiently

route information. If network nodes (users) are mobile, then static routing tables are inappropriate, and nodes require some mechanism for finding the best node to which to forward a message to, for the message to effectively reach its destination. As the destination of many messages is a node known to the source node, and many of the intermediate nodes may also be nodes known to the source (e.g., co-located nodes in a home or workplace), many researchers have explored the use of *social network* information for building opportunistic routing tables. By examining the social network of the nodes visited by a particular node, it may be possible to optimise routing by forwarding messages to nodes that are encountered more often.

To build the social networks for each node, however, data (e.g., encounters), must be collected so that social networks can be *detected* or discovered. This can lead to delays in bootstrapping the opportunistic network which may impede its effectiveness. We observe that many networked applications involve communicating with a recipient that is known to the sender, i.e., a node that is part of the sender's existing social network.

Thus, might it make sense to use the network of friends in the physical world to determine forwarding paths in a mobile network? Such a social network is already known to a node, and so requires little time to generate, compared to the social network collected from an encounter trace.

This chapter thus attempts to answer the following question:

1. Can we use SRSNs to improve opportunistic routing protocols?

This work was conducted with Dr Tristan Henderson, Dr Martin Bateman, Dr Saleem Bhatti, and Devan Rehunathan, who helped with practical aspects of the experiments, and originally published in [13].

4.1 Experimental Setup

The purpose of this experiment was to investigate what relationship there is between a SRSN and a DSN. So, we gathered SRSN and DSN information from a set of volunteers.

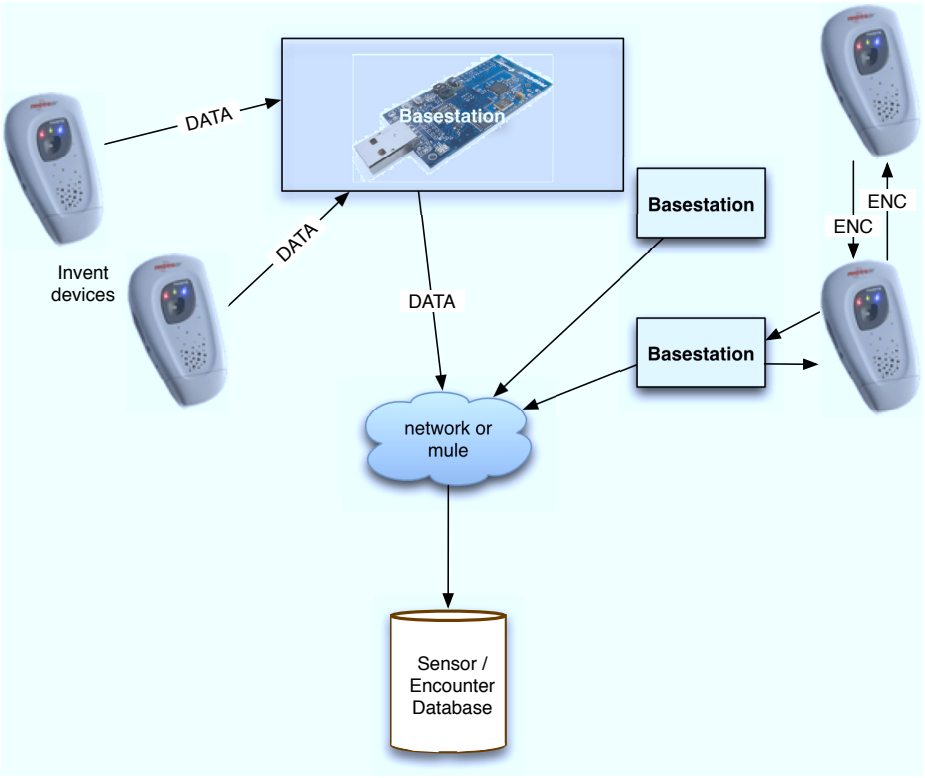


Figure 4.1: SASSY architecture. Nodes upload their detected encounters to the basestations.

We use these data to explore this relationship, to answer the question above.

We set up a mobile sensor network comprising mobile IEEE 802.15.4 sensors (T-mote Invent devices, Figure 4.1) carried by human volunteers, and Linux-based basestations that bridge the 802.15.4 sensors to the wired network. T-mote Invent devices can detect each other within a radius of ∼10m. These encounters are stored in the Invent devices and are uploaded through the basestations to a central database, as seen in Figure 4.1. The system is called SASSY (St Andrews Sensing SYstem).

For the experiments described in this chapter, we deployed 25 Invent devices amongst 20 undergraduate students, three postgraduate students and two members of staff. To upload encounters, we deployed three basestations across the two Computer Science buildings in our institution. Participants were asked to carry the devices whenever possible over a period of 79 days. We could detect Invent-to-Invent encounters anywhere throughout the town of St Andrews and beyond; they were not limited by basestation placement.

The Invent devices were programmed to broadcast a beacon every 6.67 seconds. [1] When other devices detected these beacons, they recorded a timestamp (and other information such as signal strength) for this beacon. This timestamp, the sending device's ID and the other information form a Sensor Encounter Record (SER). When these SERs are uploaded to a basestation, the basestation adds the ID of the uploading device and the basestation's ID, and records these in a central database.

To conserve storage on the Invent devices (which only have 48KB of storage space), we only recorded the maximum, minimum and mean measurements for encounters that lasted more than one polling interval.

We use the participants' *Facebook*[2] social network information to generate a topology. We consider this the SRSN. We also generate a topology using the SERs to create a social network, similar to [36]. We refer to this as the DSN.

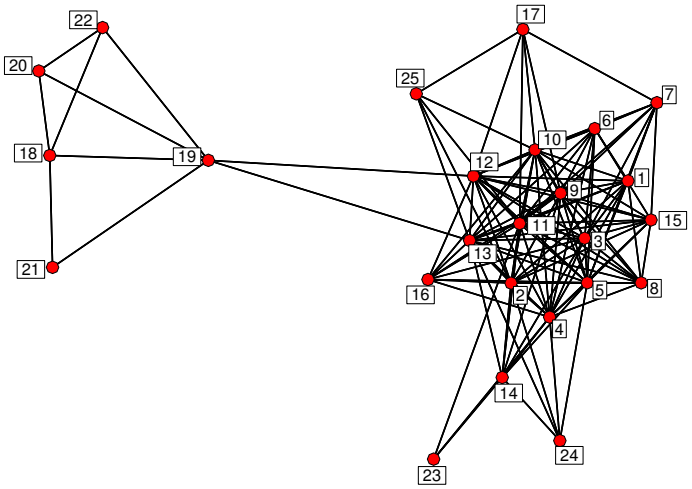
These data are referred to as the SASSY data from this point forward. We have made these data available online for other researchers to use [14].

4.2 Self-Reported And Detected Social Networks

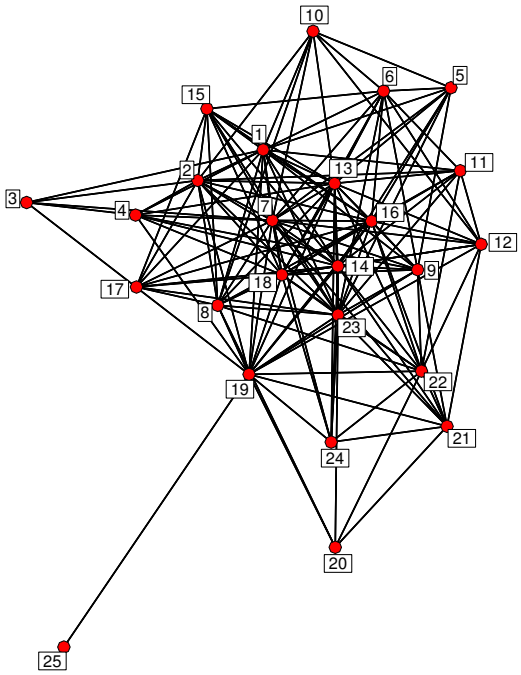
Before we can examine whether the use of SRSNs instead of DSNs has an impact on opportunistic network performance, we must answer our first question: *Are detected social networks and self-reported social networks similar?* Figure 4.2a and Figure 4.2b show the topologies for the SRSN and DSN for the SASSY data. We can observe differences between the two networks, but to better understand these differences, we employ techniques from traditional social network analysis. We may observe properties that may improve opportunistic routing.

[1] Found experimentally using two users walking each with a device. This period was determined experimentally to allow detection of encounters at walking speed, whilst preserving battery life.

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



(a) The SRSN graph. There are two groups of nodes: the small group is the staff and postgraduates, and the large group is the undergradaute student group.



(b) The DSN graph. At first glance it appears all participants bar one seem to be in the same group. On average nodes in the DSN have more ties.

Figure 4.2: The topologies of the SRSN and DSN for the SASSY trace. The node labels are consistent across both plots, i.e., node 1 in the SRSN is also node 1 in the DSN.

4.2.1 Structural equivalence

The notion of *structural equivalence* allows us to compare the ties between nodes (or *actors* in social network analysis terminology) within social networks [22]. Actors who have identical relationship ties to the same group of actors are *structurally equivalent* and are referred to as being in the same *equivalence class*.

To calculate structural equivalence we create a matrix of the ties between actors (a *sociomatrix*); if an actor i has a social tie to actor j, then the element (i, j) has a value of 1; otherwise the value is 0. If actors i and j are structurally equivalent, the entries in their respective rows and columns of the sociomatrix will be identical (i.e., the Euclidean distance between them is 0). By computing distances between all n actors in the network, we create an $n \times n$ matrix that shows the structural equivalence of each actor.

Using these Euclidean distances as a metric, we can plot *dendrograms* for the SRSN (Figure 4.3a) and DSN (Figure 4.3b). Dendrograms can be used to understand clustering; we say the nodes are clustered, where each mutual cluster is a set of nodes whose largest intra-group distance is smaller than the distance to the nearest point outside the set. For this work, the important feature to note is that nodes on the same 'branch' of the dendrogram are considered to have shorter distances between them, and are thus said to be clustered.

4.2.2 Role equivalence

Closely related to the concept of structural equivalence is *role equivalence*. This allows us to examine clusters in a social network and also compare the clusters between different social networks. Two actors i and j are role equivalent if the collection of ways in which i relates to other actors is the same as the collection of ways in which j relates to other actors [143].

To examine role equivalence graphically, we use *blockmodels* following [145]. The blockmodel can be thought of as a graphical abstraction over the sets of ties between nodes, and the function to produce it takes the clustered nodes from the dendrogram as input. Each block in the blockmodel indicates whether or not the column actor can be thought of as

having equivalent ties to other nodes as the row actor. A position is assigned to each actor in the network. For each pair of structural equivalence relationships (node ties), we determine whether a tie exists between the positions of the relationships, i.e., do the ties from actor j match those of actor i? If there is sufficient overlap of ties to satisfy the equivalence criteria (in our case: is there at least one match in every row and column of i and j's role sets), then a block is added to the blockmodel diagram in the jth column for the ith row. We must however, pick the number of output roles we expect to see. The blockmodelling algorithm will divide the nodes into the number of roles that the administrator expects to see.

4.3 Datasets

To show that the results are not peculiar to the trace data we collected, all analysis is also performed on two other traces. This section describes the datasets that are used to drive simulations. These are all available on the data hosting website CRAWDAD[3], which provides datasets to researchers.

4.3.1 HOPE

The HOPE dataset [1] traces the movements of 767 attendees at the seventh Hackers On Planet Earth conference. Participants registered their interest in topics and specific sessions before attending the conference. On arrival they were given an RFID tag to carry with them, these tags were then detected by RFID readers distributed geographically around the conference facilities.

SRSN

The self-reported social network needs to be extracted from the raw data provided. These data contain the nodes' declared interest in up to five of eleven topics. We treat all indi-

[3] http://crawdad.org

viduals claiming to have an interest in the same topic as a clique. We do this for all interests and connect the cliques together, and the resultant graph is the SRSN.

DSN

To create the detected social network, contacts between individuals need to be determined from the following raw data files: attendance of talks, detection of individuals at specific readers on a continuous timescale, and normalised time spent at locations over 5 minutes blocks. We assume that nodes attending the same talk are assumed to be in contact.

The static RFID readers logged detection of mobile nodes, with the information stored in the format <timestamp, nodeid, location>. This provides the order that nodes visited locations. As the time is represented as a timestamp, we need to compute the length of time nodes are present at a location before we can assume co-location. A subset of the data is of "normalised node location". This provides the name of the locations a node was detected at in a five minute window, with a value in the range $0 - 1$ which represents the normalised length of time the node was present at this location. This allows us to compute the amount of time a node was at a location. By combining both of these sets of data we can work out when, and for how long, nodes visited specific locations at the conference. We can determine if nodes were co-located in space and time; and, if so, we can infer that an encounter took place.

As RFID-reader detected encounters required more calculations, and therefore increased likelihood of error, the talk attendance data are assumed to take precedence over the interpolated encounters between individuals in the event that nodes appear to be in two places at once.

4.3.2 Reality Mining

In the RM trace from MIT [40], 99 individuals were given a mobile phone to carry that used Bluetooth to detect co-location. 75 of the users were either students or faculty at the laboratory, while the remaining twenty-five were incoming students at the adjacent

Table 4.1: Dataset graph statistics

	Trace					
	SASSY		RM		HOPE	
Property	SRSN	DSN	SRSN	DSN	SRSN	DSN
Number of Vertices	25	25	97	96	414	410
Number of Edges	127	155	107	2328	67836	80183
Clustering Coefficient	0.748	0.672	0.255	0.702	0.834	0.976
Graph Density	0.406	0.496	0.023	0.505	0.792	0.954
Graph Components	1	1	28	1	1	1

Graph density is the ratio of the number of edges in the graph over the maximal number of edges($N(N-1)$).

Table 4.2: Trace properties

Trace	SRSN	DSN
SASSY	Facebook Friends	ZigBee Encounters
HOPE	Topic Interest	RFID Co-Location
RM	Phone Contact List	Bluetooth Encounters

business school. The authors logged cell-tower ID to determine approximate location and at the same time logged Bluetooth devices.

The information collected included call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle). Participants carried the devices for nine months. We use the user-programmed phone book as the SRSN, and the Bluetooth encounters between participants as the DSN. We can see the details of the input data used to create the SRSN and DSNs for all traces in table 4.2.

4.4 Network Analysis

We now look at the social-network analysis techniques described in section 4.2.1 and 4.2.2 to the traces found in section 4.3. We create dendrograms and blockmodels for each of the datasets using R.

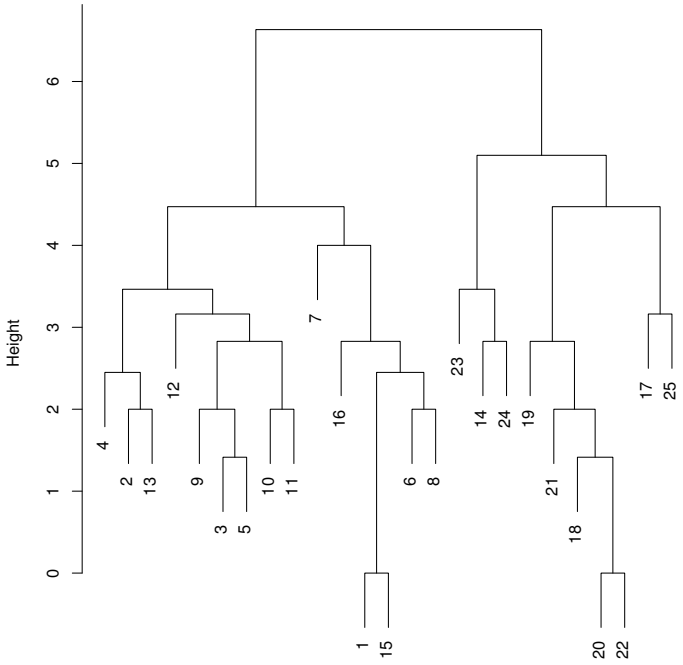
The table in 4.1 shows the graph details of the SRSN and DSN for each trace. We see that for each trace, the DSN is always denser than the SRSN, meaning that we expect that more paths will be available for routing. We would expect this, because data from the SRSN imply a social connection of some kind, which we would expect to be stronger than the social connection that arises from two individuals being co-located.

The edges are not weighted for either the DSN or SRSN. Many of the traces do not include multiple interaction information for the nodes. For example, the SASSY trace includes a single connection between each of the SRSN nodes, similarly in the HOPE trace. We therefore have no measure of the relative importance of a tie in each node's SRSN. With the DSNs, edge weightings could, for example, be determined using a measure of frequency of encounter or length of encounters. This approach, however, would result in edge weights linked strongly to the encounter sampling methodology, which, in turn may affect the graph level analysis. To allow us to perform a fairer comparison of opportunistic network performance, we therefore do not weight edges for any of the traces.

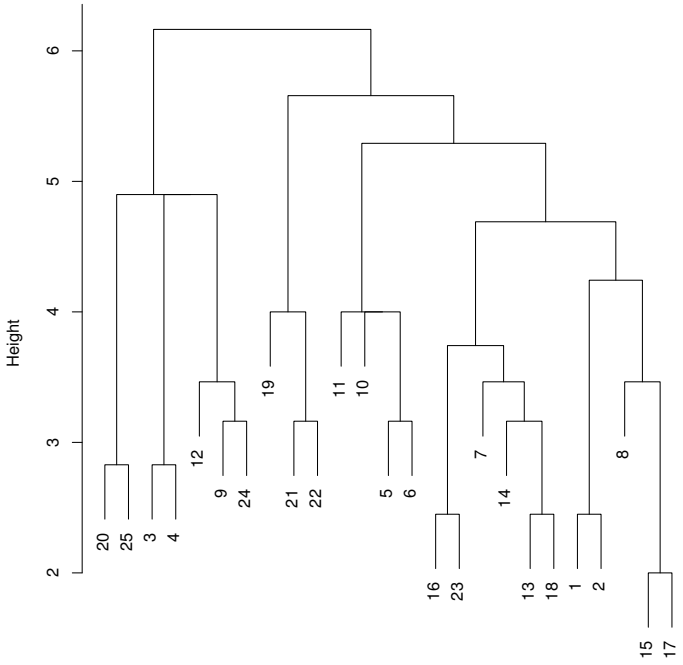
4.4.1 SASSY

When analysing the SRSN blockmodels we see four roles, each of which can be seen as a cluster in the dendrogram Figure 4.3a, or as a section of the blockmodel (Figure 4.4a).

Nodes 4, 2, 13, 12, 9, 3, 5, 10, 11, 7, 16, 1, 15, 6, 8 form role 1: These nodes are almost entirely connected to themselves. These nodes are the most popular, with lots of ties; and ties to the widest variety of nodes. In Figure 4.2a, these nodes are located close to the centre. These nodes represent a *clique* — they all have ties to each other, but there are no other nodes that are directly connected to all the members of the group [143].

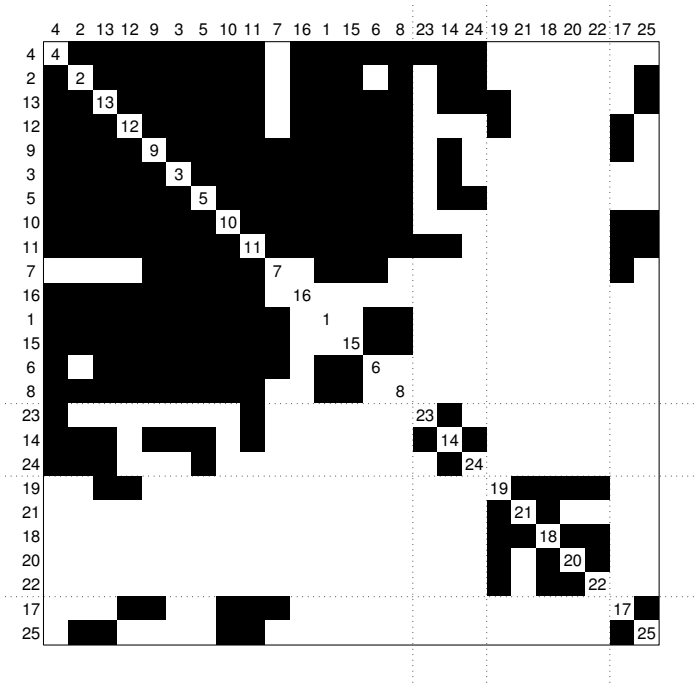


(a) The SASSY SRSN dendrogram. Here we see four clusters. Only four nodes are completely similar to one another.

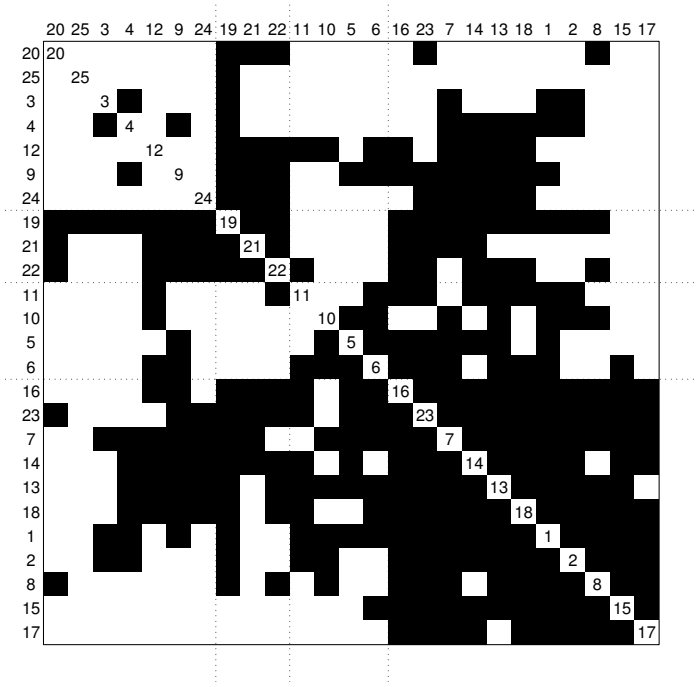


(b) The SASSY DSN dendrogram. Here we see four clusters. No nodes are completely equivalent, meaning that there are no perfect alternate nodes that have matching social networks.

Figure 4.3: The equivalence clustering Euclidean distance dendrograms. Height indicates the Euclidean distance.



(a) SASSY SRSN blockmodel. Nodes in the large group are similar to one another, and form the bulk of the nodes. The outlying staff and postgrad role is connected to this main group via node 19. The other two roles are made up of outliers from the main group.



(b) SASSY DSN blockmodel. We see four roles, note that the first role's (upper left quadrant) defining feature is that nodes are not similar to one another, while the fourth role's (bottom right quadrant) is that nodes are almost completely similar and interchangeable with regard to connecting to other nodes in the same role.

Figure 4.4: Blockmodels of role equivalence for the SRSN and DSN. Dotted lines indicate role divisions.

We would expect that nodes in this group would be able to communicate with each other with low delay and high probability of delivery success.

Nodes 23, 14, 24 form role 2: Three nodes on the edge of the main group. They are found in the middle of the dendrogram (Figure 4.3a). They are slightly similar to each other, but are grouped together as their common property is a fewer number of connections than nodes in role 1.

Nodes 19, 21, 18, 20, 22 form role 3. These nodes form the postgraduate and staff group. They are similar to each other, and connect to the other nodes only through node 19. We would imagine then, that node 19 is an important node for maintaining routing paths.

Nodes 17, 25 form role 4. These nodes are again outliers from role 1. They are on the other side of role 1 to role 2, and are therefore not grouped together.

In contrast with the SRSN, we found four weakly-defined roles in the DSN blockmodel (Figure 4.4b).

Nodes 20, 25, 3, 4, 12, 9, 24 form role 1. There nodes are on the edge of the graph (Figure 4.2b), and while they are not similar to each other they are grouped together, as they have the fewest connections that are useful to other nodes.

Nodes 19, 21, 22 form the second cluster and role 2. These nodes have connections to most of the nodes in roles 1, 2 and 4, and only a few connections to role 3. Crucially, they provide links to the more central nodes for those in the outlying cluster, and hence do not fall into the outlying cluster themselves.

Nodes 11, 10, 5, 6 form role 3. They lie near to (but not in) the centre of the network. They can also be considered edge nodes, but they have more connections into the centre than role 1, and are also highly connected to role 1, yet hardly similar to role 2, and therefore are not grouped with role 2.

Nodes 16, 23, 7, 14, 13, 18, 1, 2, 8, 15, 17 form role 4. This is the most central role. The nodes are very similar to one another, and share many similarities with nodes in all the other roles.

On average, nodes in the DSN have a greater number of ties than in the SRSN. In both cases the roles indicated in the blockmodel confirm the clusters described in the respective dendrograms, and also help us to distinguish the roles more clearly. The roles are less well defined in the DSN, as the blockmodel does not show as obvious divisions as in the SRSN. The SRSNs roles seem to form in groups of nodes with similar relations to each other and with clear group boundaries. It is also noticable from the graph that in an opportunistic network roles 2, 3, and 4 might heavily rely on role 1 to provide paths in the network. In the DSN, however, divisions seem to be distinguished by number of ties to the center of the network.

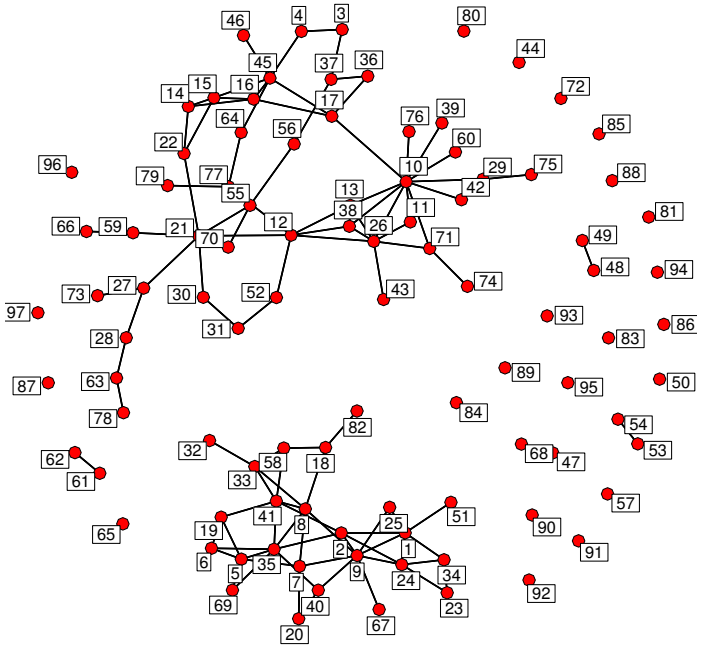
The similarities that these analyses show between nodes allow us to make predictions about which nodes will be the most useful for forwarding. Nodes that are similar to one another can be viewed as providing similar forwarding paths. Routing protocols may be exploit this in the event expected nodes are not present.

This is a feature of the blockmodel that would not have been obvious from simply inspecting the topology diagrams. Blockmodels may therefore provide information useful in opportunistically routing messages.

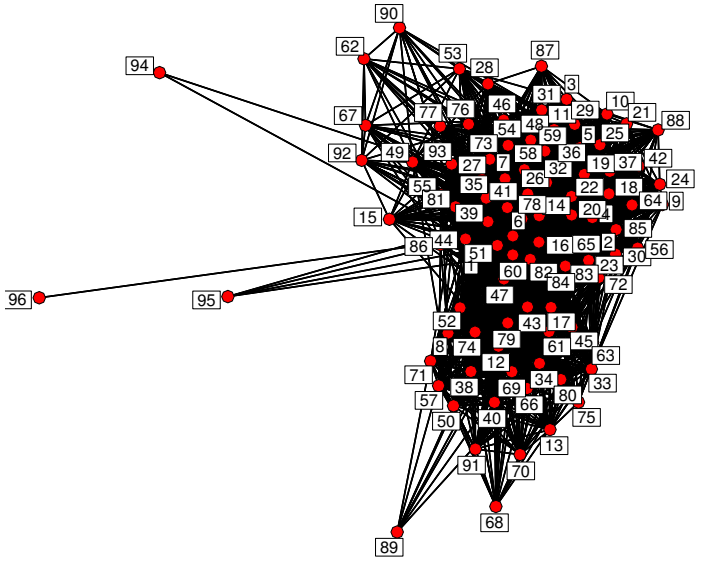
Reality Mining

When we compare the RM node diagrams Figures 4.5a and 4.5b we notice that the DSN has one large component, while the SRSN has 28 components. When trying to route messages the nodes have more encounters than expected social contacts (SRSN neighbours) and therefore have more options available to them for routing.

In cluster dendrograms Figures 4.6a and 4.6b, we see that in the SRSN trace 35 nodes have a difference of 0 with other nodes, meaning these nodes' neighbours are exactly the same as at least one of the other nodes. This is not the case for the DSN. Nodes on average have a difference of between two and eight. Overall there is less difference between nodes in the SRSN than the DSN. The maximum difference on the y axis is just over six in the SRSN, while is is almost 14 in the DSN.

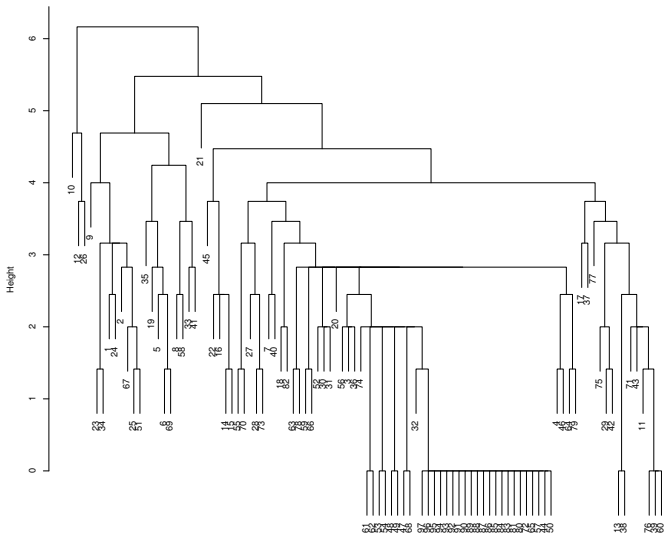


(a) Reality SRSN graph. Here we see two large groups, with several individuals and 4 pairs. In total there are 28 components.

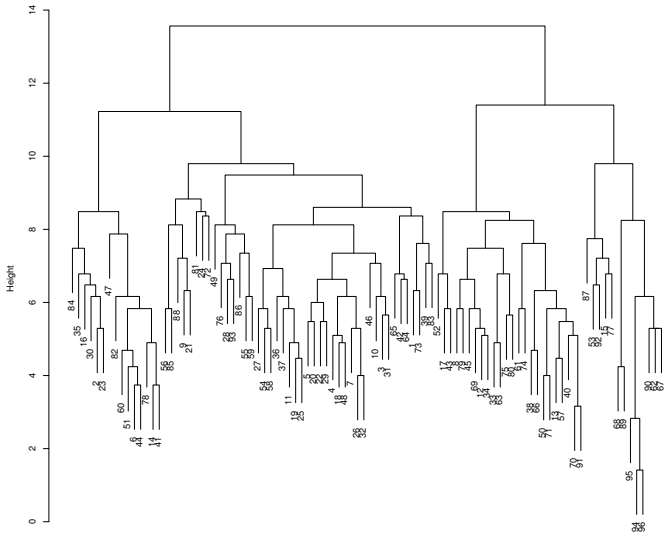


(b) Reality mining DSN graph. Here we see one component. Not all nodes, however, have a large number of contacts.

Figure 4.5: SRSN and DSN graphs for the RM trace.

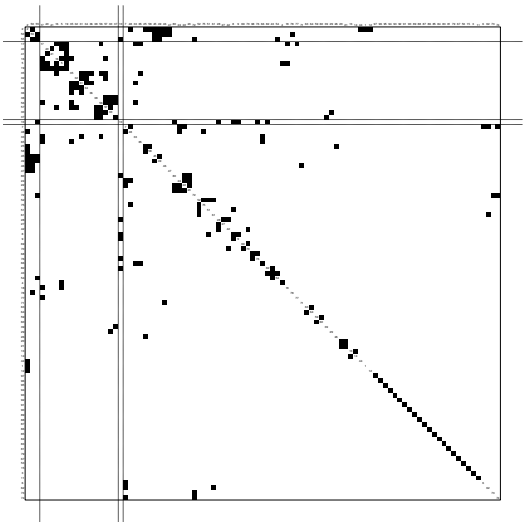


(a) Reality SRSN cluster dendrogram. We see a large difference between the nodes, as expected from inspecting the SRSN graph. There appears to be three main clusters. The outliers are all very similar, and are clustered together. High betweenness nodes in the large component: 10, 12, 26 are clustered together also.

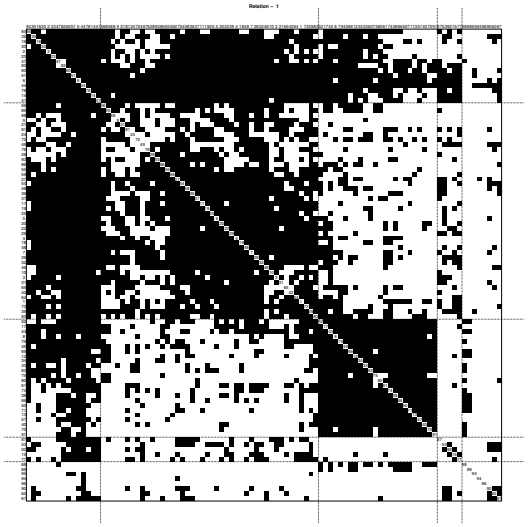


(b) Reality Mining DSN cluster dendrogram. There appears to be 5 large clusters. The middle branch there is little difference between the nodes. None have a difference of 0.

Figure 4.6: Reality Mining Cluster Dendrograms. Height indicates Euclidean distance.

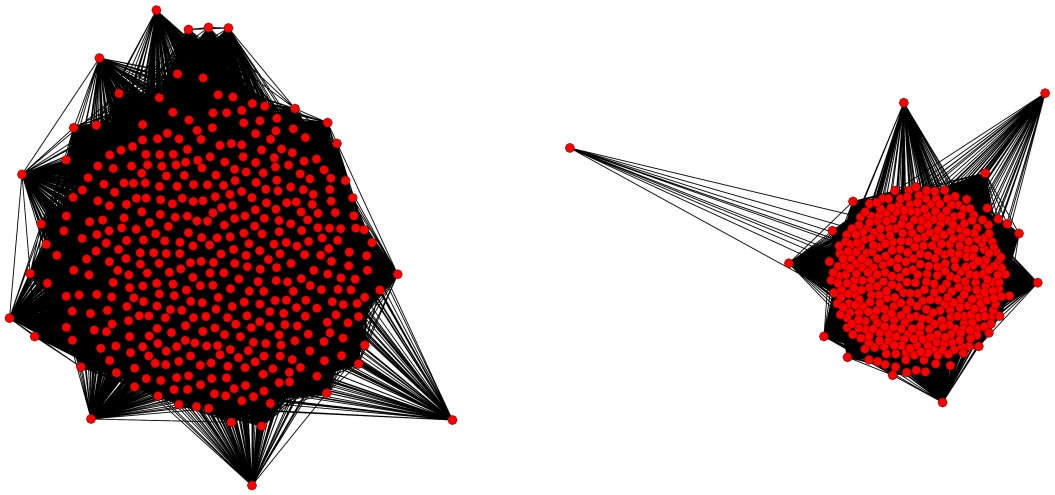


(a) Reality mining SRSN blockmodel. Here we see very little similarity between the nodes in the SRSN, this is most likely because of the small number of connections between nodes. There are less structurally equivalent nodes than in the DSN.



(b) Reality mining DSN blockmodel. We see four different groups in the network. Nodes in the dark square region in the top left are all very similar to one another, and are frequently similar to nodes in two of the other groups. Nodes in the second group (56...83) are near the middle of the DSN graph, and have are similar to themselves and nodes in group 1. Nodes in the third group are slightly similar to nodes in the first group, but are very similar to one another. They are found in the group in the bottom right of the network graph. Nodes in the fourth group are neither similar to nodes in any of the other groups, nor to nodes in their own group.

Figure 4.7: Reality Mining Blockmodels. Nodes in the SRSN show less similarity with one another than the DSN nodes.



(a) Hope SRSN graph.

(b) Hope DSN graph.

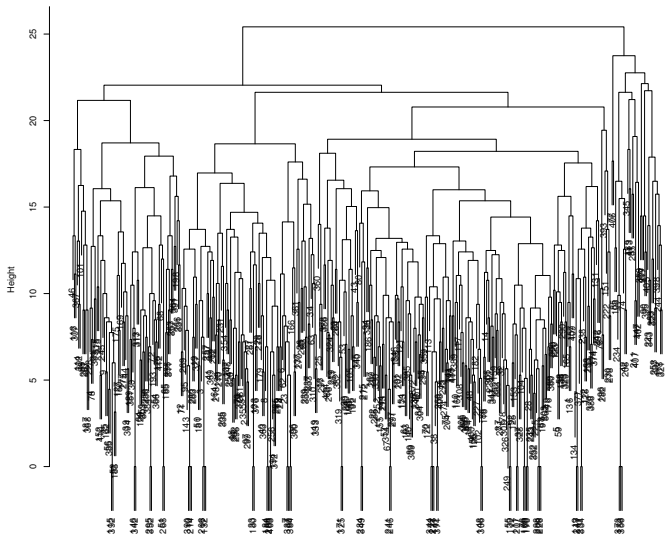
Figure 4.8: The SRSN and DSN graphs for the HOPE trace. Both are highly connected dense graphs.

The RM blockmodels show that there is little similarity in the SRSN Figure 4.7a, despite the number of nodes with a distance of zero to other nodes. The DSN Figure 4.7b shows two groups (84...85, 52...91) with high similarity within each group, yet in the cluster diagram they have distances greater than 0. These graphs show that just using one technique (clustering or blockmodelling) is not enough to capture potential node similarity.

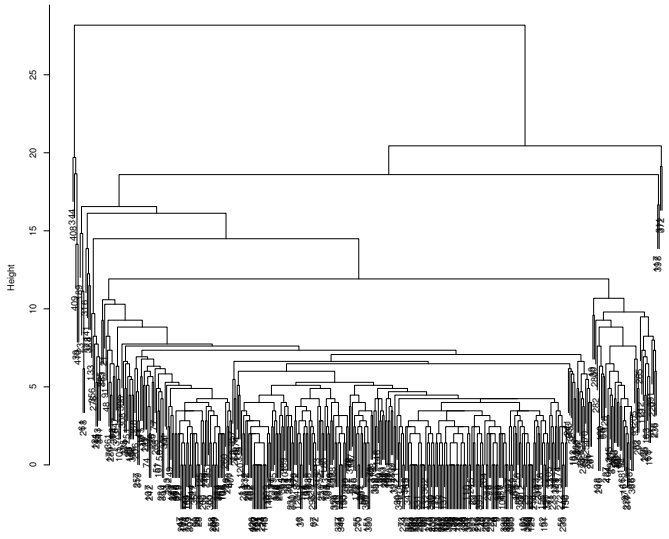
HOPE

The SRSN and DSN (Figures 4.8a and 4.8b) for the HOPE trace are both very dense (0.792 and 0.954 respectively), yet when we look at the clustering dendrograms (Figures 4.9a and 4.9b) we see that far more nodes in the DSN have a difference of 0 in comparison to the SRSN. Due to density of figures, these dendrograms are included for illustrative purposes only. The DSN dendrogram shows that in the DSN, nodes have a wider range in difference than in the SRSN. Despite this wider range in difference, overall there is a lower average distance in the DSN that the SRSN.

The blockmodels for the SRSN (Figure 4.10a) and DSN (Figure 4.10b) show nodes in the DSN are very similar. The SRSN shows some nodes with high similarity, representing

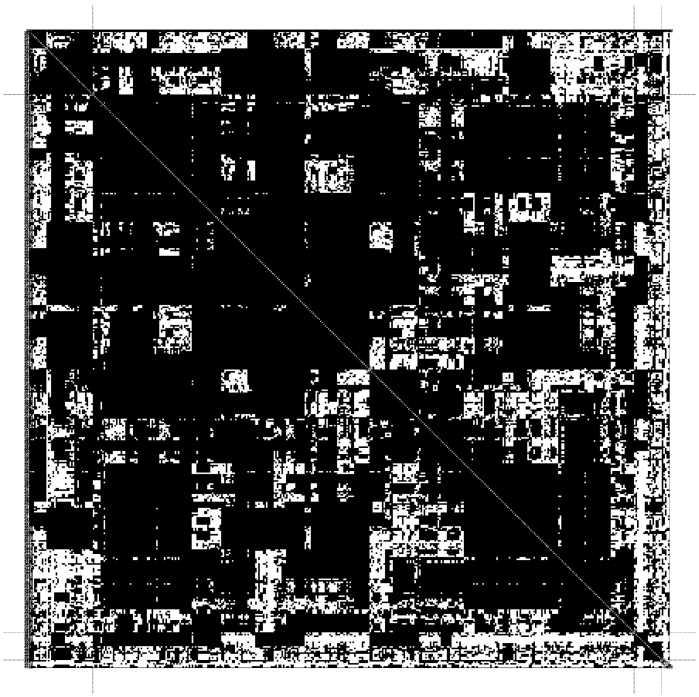


(a) Hope SRSN cluster dendrogram. We observe six main clusters, all but one containing nodes with a distance of 0. The cluster on the right hand side from the root is made of the edge nodes in the graph.

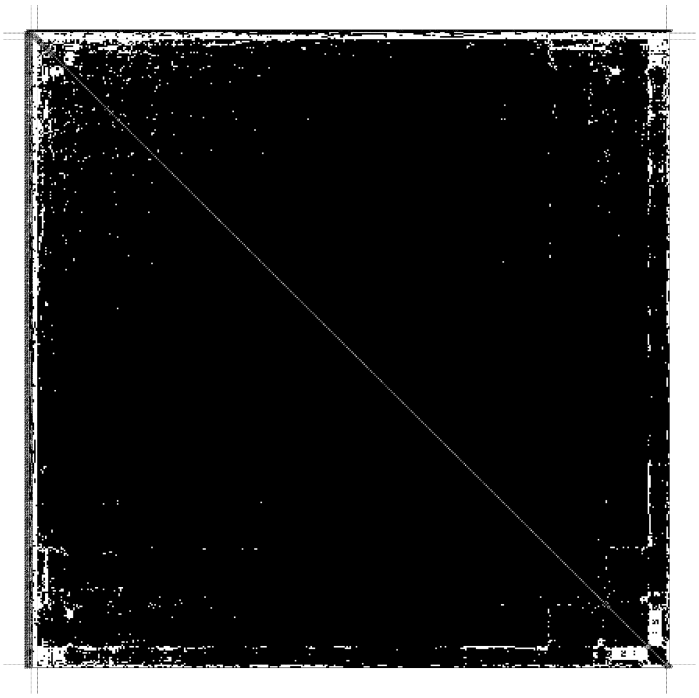


(b) Hope DSN cluster dendrogram. We observe that the very large cluster has many nodes with distance of 0. This is in contrast toe the SRSN which has fewer nodes with distance 0.

Figure 4.9: HOPE Cluster Dendrograms. Due to size, figures only included for illustrative purposes.



(a) Hope SRSN blockmodel. We see several groups with similarity in connections. These may represent cliques within the graph.



(b) Hope DSN Blockmodel. We can see that the nodes in the DSN are very similar to one another. The graph is almost a single clique.

Figure 4.10: HOPE Blockmodels.

cliques of nodes within the large group itself.

We have seen that blockmodels and dendrograms offer a way to see node similarity and provide a numeric value for how different the nodes are to each other. It is, however, required that we manually decide how many different groups/roles there are in the networks via inspection of the dendrogram/blockmodel.

We have seen that the SRSN and DSNs differ in structural and role equivalence even when we compare two network graphs that are very dense and appear similar in structure. We now go on to see how these differences affect opportunistic routing.

4.5 Hypotheses

Nodes in the DSNs have a greater number of ties than in the SRSNs, and these ties are more frequently to nodes outside their role. We therefore infer that the nodes in the DSN may be more easily reached. For example, we see in Figure 4.2a and Figure 4.4a that there are fewer paths and more clearly defined roles than the DSN equivalents Figure 4.2b and Figure 4.4b.

We use similar metrics to those listed in [71], namely:

- Delivery ratio — the number of messages that have been delivered divided by the total number of messages created.
- Delivery cost — the number of medium accesses, divided by the number of messages delivered.

We make the following hypotheses:

1. An opportunistic network application using a SRSN as a routing table would have a lower delivery ratio than one using a DSN, as there are fewer links available, and nodes are more restricted in the ways they relate to each other.
2. An opportunistic network application using a SRSN as a routing table would have a

lower delivery cost than one using a DSN, as there are fewer links available, and so nodes are likely to generate fewer transmissions.

4.6 Experimental Results

To test the hypotheses of Section 4.5, we use the SRSN and DSN as inputs to a simulated opportunistic network and calculate delivery ratio and delivery cost metrics to determine message-passing performance.

4.6.1 Simulation setup

An important factor in the design of a opportunistic network is the amount of time that it takes to pass a message from one node to another – the *bundle-transfer time*. If messages are large and take a long time to transfer between nodes, then we are reliant on nodes encountering each other and staying within range of each other for a long time in order for the opportunistic network to be effective. We assume a bundle-transfer time of 30 seconds, which we believe to be a reasonable minimum encounter time[4]. Experimentally, we also found that, for our experiments, beyond 30 seconds, the delivery ratio becomes negligible as there are insufficient encounters longer than 30 seconds.

We hold this bundle-transfer time constant, and experiment with different time to live (TTL) values: this allows us to determine opportunistic network performance under different application scenarios. For instance, a disaster scenario application might require a low message TTL, whereas a holiday maker sending photos to their friends may not worry about timely delivery and so will be content with a high TTL.

The encounter trace data were used to create a trace-driven simulation. Simulation parameters can be seen in Table 4.3. Randomly selected nodes send messages to randomly selected destinations, where the total number of messages created in each simulated day is the number shown in the table. To increase realism, all random values are taken from an exponential distribution.

[4]Computed via experimentation with N95 smartphones sending 1MB file

The messages are sent in an unicast fashion, rather than a one-to-many or a many-to-many approach. This is to capture an application similar to a mobile network short-messaging-service. A one-to-many messaging approach is more appropriate for advertising than messaging. Many existing works similarly use an unicast approach for their simulations [35,36,72,141].

For this experiment, devices are assumed to have infinite buffers and energy. This is to minimise the effects from including these parameters, and to ensure that any divergence arising is from the difference between the protocols, rather than external factors.

Table 4.3: Simulation parameters

		TTL values	
Trace	Messages per Day	Hours	Days
SASSY	20	†	1, 2, 3, 4, 5, 8, 10, 20, 30, 40, 50, 80
RM	84	†	1, 2, 3, 4, 5, 8, 10, 20, 30, 40, 50, 80
HOPE	414	1, 2, 3, 4, 12	1

† No runs with these units

The traces are initially parsed to generate the SRSNs and DSNs for nodes. Then, traversing the encounters, a message is passed to the encountered node if the encountered node is in the source's SRSN or DSN, respectively. This is so that the DSN protocol does not flood the network in dense traces. We condense the encounters as follows: we assume bundles of messages to be passed within 30 seconds, so any encounters shorter than this period of time are not included in the simulations.

It is important to note that the DSN is not growing during the course of the simulation; it is static and is calculated at the start of the simulation. This means that the DSN nodes are able to use future information. They use knowledge of encounters that have not taken place yet. This will likely result in the DSN protocol having a higher performance than in a deployment where future information is not available.

4.6.2 Simulator

We use a custom discrete-event simulator, written in Python to perform the simulations of opportunistic routing. A custom simulator was more appropriate than an existing simulator such as NS-2, as we do not need to take mobility into account. [5] The simulator assumed homogeneous devices, where all nodes have equivalent buffer and battery capacity.

First, the simulator parses the input parameters, for example, reading the number of messages to be sent per unit time. Second, the simulator generates the initial copies of the messages and places them in the nodes message buffer.

Each trace that we use presents a set of events (the DSN) and an SRSN. Assuming the trace is sorted in ascending timestamp order, the simulator steps through the events exchanging messages between devices according to the opportunistic-routing protocol in use.

The simulator was validated by performing a simulation of SimBet [36] and Epidemic routing according to the authors parameters [36] and achieving equivalent results.

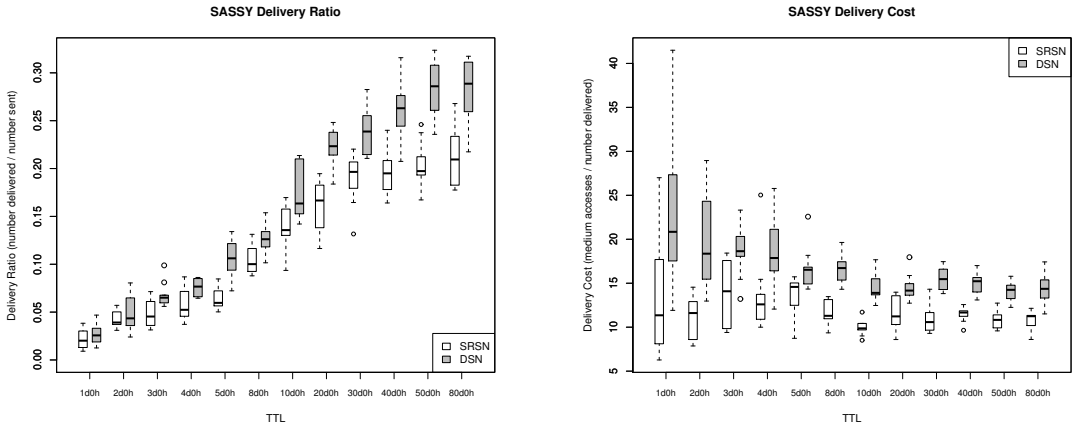
4.6.3 Results

Delivery ratio

SASSY Figure 4.11a indicates similar trends for both DSN and SRSN. As might be expected, the delivery ratio increases along with the TTL. Both protocols result in somewhat low delivery ratios overall, with a maximum of 33% of the messages being delivered. This is an artefact of our trace; it turns out that the participants in our experiment did not encounter each other frequently enough to create a very effective opportunistic network.

The DSN results in a slightly higher delivery ratio than the SRSN (at most 5%). This difference was found to be statistically significant using a t-test ($p < 0.01$).

[5] http://isi.edu/nsnam/ns



(a) Statistically significant difference ($p = 0.6 \times 10^{-7}$), due to the divergence in ratio at higher TTL values.

(b) Significant difference ($p = 4.14 \times 10^{-10}$). The cost is higher at the beginning of the simulation, as there are less deliveries per medium access.

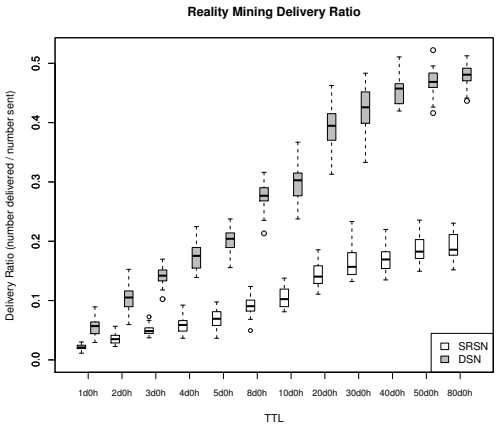
Figure 4.11: Results for the SRSN and DSN for the SASSY trace.

Reality Mining In Figure 4.12a we see that there is a large difference between the delivery ratio for DSN and the SRSN. This is likely because nodes in the DSN have more contacts. The SRSN is too restrictive in the number of encounters it allows nodes to use for forwarding. Even at high TTL values the DSN does not achieve an average delivery ratio of above 0.5.

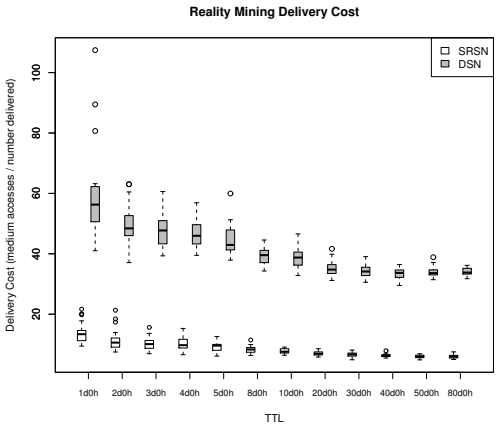
HOPE Figure 4.13a shows the HOPE delivery ratio results. We see that the difference in delivery ratio between the SRSN and DSN is not statistically significant. Both protocols achieve a high delivery ratio, at high TTLs the delivery ratios are over 0.8.

Delivery cost

SASSY Figure 4.11b shows the delivery costs for the SRSN and DSN simulations. We see that the SRSN results in a dramatically lower delivery cost, with a maximum of 73 medium accesses per message for the SRSN compared to a maximum of 187 for the DSN. Like the delivery ratio, the difference in delivery cost is found to be statistically significant via a

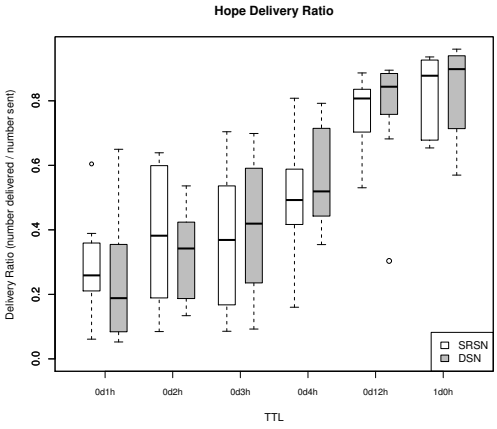


(a) Significant difference ($p = 3.10 \times 10^{-08}$)

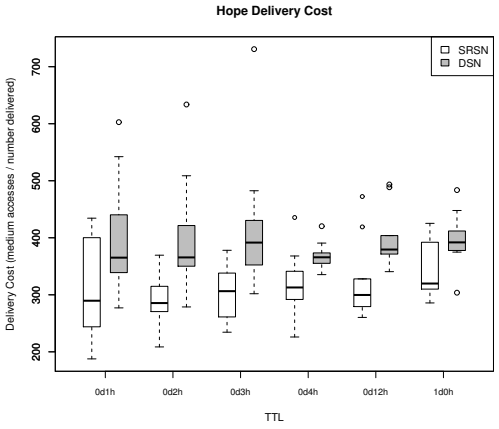


(b) Significant difference ($p =< 2 \times 10^{-16}$)

Figure 4.12: Results for the SRSN and DSN for the RM trace.



(a) No significant difference ($p = 0.6213$).



(b) Significant difference ($p = 0.00779$).

Figure 4.13: Results for SRSN and DSN for the HOPE trace.

t-test ($p < 0.01$). This is because the SRSN exploits fewer links: it forces the intermediate nodes to ignore encounters not in the source social networks that may prove useful, and provides links to intermediate nodes they may never meet.

Reality Mining The RM delivery cost shows a significant difference between SRSN and DSN Figure 4.12b. This is due to the increased number of encounters used for passing messages in the DSN simulation.

HOPE We see a significant difference between the SRSN and DSN delivery cost for the HOPE trace Figure 4.13b. These results show that even with two high density traces that look similar we may see different forwarding behaviour.

4.6.4 Discussion

We have seen that across traces which differ greatly in terms of number of nodes and density, that SRSNs can provide adequate tables to drive opportunistic routing. The SRSN performs better than the DSN (equivalent delivery ratio and lower delivery cost) in two of the three trace driven simulations.

The DSN protocol in our simulation uses optimal information, because it is calculated at the start of the simulation. In a real deployment the DSN would grow over the course of the experiment, rather than starting with all future encounters included. The number of links paths used, therefore, is inflated compared to reality.

The simulation does not require that the sources only send messages to nodes in their social network. This is because many of the nodes in SRSNs for the RM project have a small number of connections. They could skew any results, resulting in a higher standard deviation in delivery ratios. In practise, however, this may be a fair assumption.

The DSN has a higher density and one component in every trace, and in the SASSY and RM traces, the SRSN is different to the DSN (more components and lower density). The HOPE trace SRSN and DSN may appear similar when observing the graph, however,

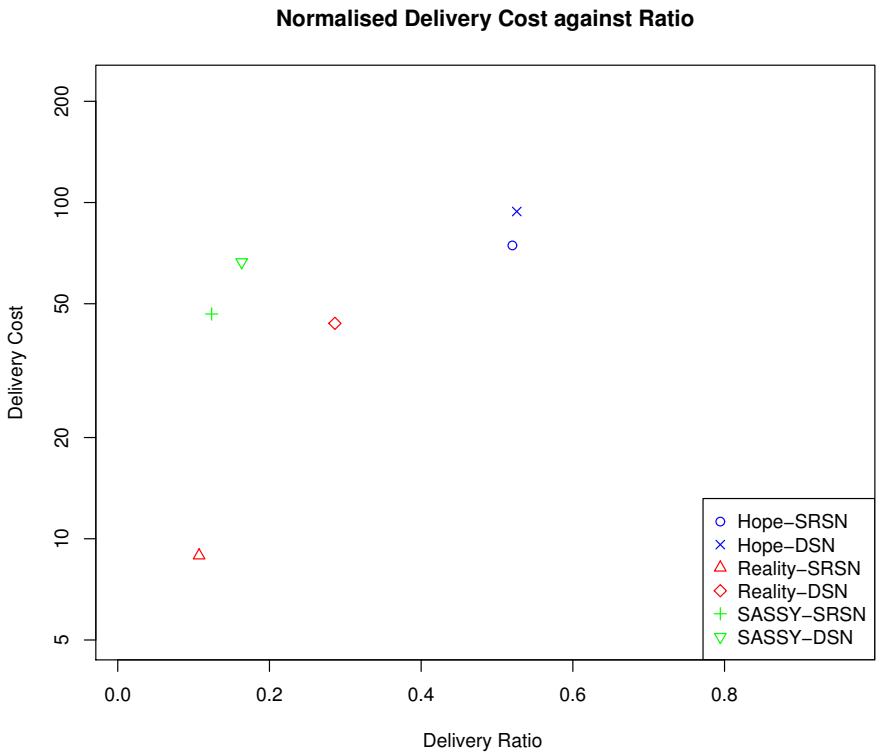


Figure 4.14: Plot of the mean normalised sending cost and delivery cost against delivery ratio. We see that in all cases the DSN has a higher Delivery Ratio and Delivery Cost than the SRSN. We know, however, that for the SASSY and HOPE traces the SRSN and DSN delivery ratios are nearly equivalent.

the social network analysis reveals differing structural and role equivalence for the nodes. This difference is confirmed in the simulation where the SRSN outperforms the DSN. The RM trace, however, shows the DSN to have a higher delivery cost and higher delivery ratio. This is likely to be because the DSN is far denser than the SRSN. For this trace we need to decide whether the cost saving of the SRSN is more important than the increased delivery ratio of the DSN. This is confirmed in Figure 4.14 where we see that the DSN has a higher delivery cost and delivery ratio. Thus, when picking a routing protocol we need to decide which is more important: delivery cost or delivery ratio, and pick the SRSN accordingly.

4.7 Conclusion

In this chapter we have examined the use of self-reported social networks for opportunistic routing. We have analysed SRSN and DSNs to show the structural and role equivalence, and used this analysis to motivate the use of SRSNs as well as DSNs for opportunistic routing. We have seen that the SRSN outperforms the DSN for driving opportunistic routing in certain scenarios. When the SRSN does not outperform the DSN, rather than performing worse, it merely displays different properties of cost and ratio. We have seen that it is the case that network operators/application designers need to consider the desired delivery cost and delivery ratio of their application when choosing a routing protocol.

We have seen that SRSNs are useful in a simple routing protocol. This may be the case because using a SRSN for routing does not require maintenance of an encounter history like using a DSN does, resulting in lower energy and memory consumption.

As we are using a trace we can scan for future information (encounters that occur after the current point in time). Whilst this use of future information is not possible in real-world deployments, a SRSN can be collected before network startup. We explore the use of SRSN data from network startup in the next chapter.

Currently, the application designer must decide on how many roles are present in the network. An automated way to choose roles would allow them to be used in routing protocols, where they may be useful. We explore this in the next chapter.

Chapter 5

Social Roles for Opportunistic Forwarding

We have seen in Chapter 4 that SRSNs can provide information that is useful in opportunistic routing. In this chapter we see examples of where existing routing protocols perform badly, and how we can improve upon existing routing protocols by using SRSN information.

This work was conducted with Dr Tristan Henderson and published in [11].

Many researchers have looked at conducting social network analysis of the encounters between mobile nodes to create routing protocols for opportunistic networks, exploiting the fact that a node that has encountered a node in the past may be likely to encounter this node in the future. For example, PROPHET [93], SimbetTS routing protocol [35] and MaxProp [19] all use encounter histories to drive routing decisions. Such routing protocols may suffer from two problems:

1. First, those nodes that are best-connected in the social network (i.e., those nodes with the highest number of "friends") might be selected very frequently for forwarding messages. This may lead to those nodes running out of energy or storage. As we saw in Chapter 3.1, as much as 63% of traffic is routed through only 10% of nodes when using protocols which rely on only the most preferred nodes for forwarding [147].

It may therefore be beneficial to consider alternative nodes for routing, rather than exclusively those most obviously suitable.

2. Second, collecting encounter information on which to analyse social networks may require time and a mechanism for aggregating these collected data. This leaves a window of time between network start-up and the point at which the protocol has built a model reliable enough to make accurate inferences for routing. During this period nodes may experience sub-optimal routing behaviour, reducing network performance. Using an alternative mechanism for bootstrapping the network before the encounter information can be gathered, may improve routing performance.

In this chapter we introduce a new opportunistic routing protocol, Social Role Routing (SRR), which attempts to solve these two challenges. Rather than rely upon on the social networks derived from collected encounter patterns, we use information from pre-existing social networks to drive routing decisions. To derive alternative nodes for forwarding, we determine the social *roles* in the social network to find classes of nodes that may be useful for forwarding. SRR bootstraps opportunistic networks while providing high performance, before alternative encounter-history-based routing protocols' forwarding decisions have stabilised.

5.1 Overview Of Role Forwarding

This section describes the role forwarding algorithm SRR. We show how, and why, we detect roles in social networks, and then how they can be used to make an opportunistic routing algorithm.

5.1.1 Role analysis

In society we are members of various social networks, from work relationships to sports groups to connections with our neighbours. We can use a record of these social network data to create self-reported social networks, such as our Facebook "friends", that can aid

opportunistic routing. By constraining the devices that can use a particular node as an intermediary for opportunistic message delivery to only those members of a node's SRSN, we may save energy (due to the reduced number of forwarded messages) at little expense (members of a node's SRSN may be more likely to aid message delivery).

As opportunistic networking devices are frequently assumed to run on battery power, we should consider battery usage as a cost to users. We want to minimise this cost by reducing redundant forwarding of messages. One way to do this may be to constrain the number of devices allowed to use a particular node as an intermediary for message delivery. We can use the SRSN information to do this, as we can make an assumption that users would be happy to relay messages for those individuals with which they have a pre-declared relationship. SRR exploits SRSN information in this way.

Certain nodes may be more central, or have many connections, in the encounter network and therefore be required to do a larger share of message forwarding on behalf of other devices. This may lead to the device being depleted of energy. Battery depletion may then, be more likely to affect hub nodes or nodes that bridge groups. For example, in Figure 5.1, all messages sent by nodes in group *A* to group *B* would likely pass through node 1 (as it is on the shortest path). Node 1 might therefore run out of battery quickly, or be less likely to participate in an opportunistic network if its resources were to be heavily-utilised, due to selfishness or lack of battery. We thus need to find nodes that can function similarly to node 1 (in that they connect group A to group B). In this case messages could be routed via nodes 12 and 13 in certain circumstances, which can be used as an alternative forwarding path.

As well as being members of social networks, we function in society as members of various roles. For instance, we can view a business as a network of employees, all of whom have particular roles which define their connections to the other employees. An example can be seen in Figure 5.2: assistant managers connect managers to non-managerial workers. If an assistant manager is not available to pass on work instructions to the workers, a manager may select another assistant manager to do this, as all members in the assistant manager role perform the same task (passing on instructions).

We extend this analogy to opportunistic networking. By passing messages to nodes

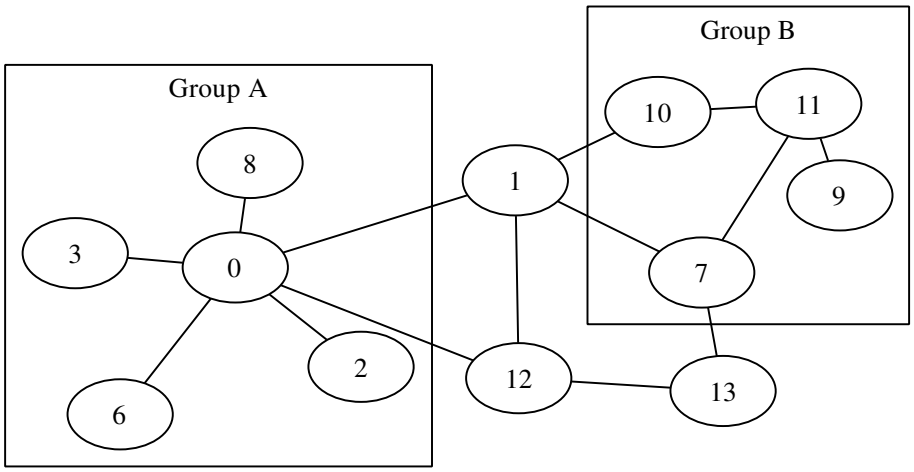


Figure 5.1: An example node contact graph. Routing messages from group A to B via the shortest path (node 1) may overburden the device's resources. Utilising an alternative path (nodes 12 and 13) may be more appropriate.

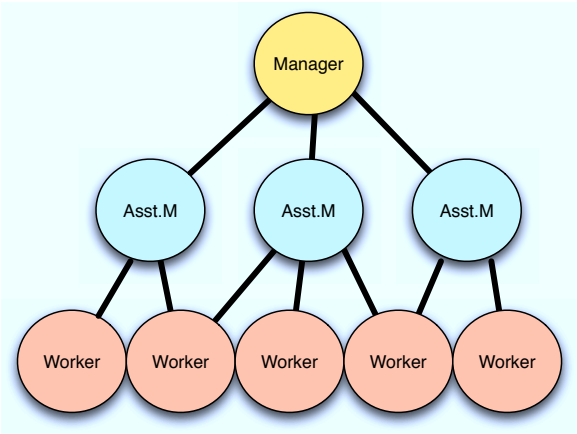


Figure 5.2: Here we see the nodes in a company network. Each colour represents one role. Managers pass instructions to Assistant Managers, who pass the instructions on to the workers. Similarly the workers pass the results to the Assistant Managers, who pass the results back up to the Manager.

performing a similar role in an opportunistic network, we can find alternative nodes for times when the preferred nodes are unavailable, or to avoid overloading the preferred nodes.

We can consider roles useful for routing because:

1. Nodes can be in the same role as nodes they have not encountered
    - This provides greater resilience of roles in comparison to grouping nodes with their neighbours. If there is a network partition, groups are more likely to lose connectivity than roles, because members of groups are located together in the network topology. Members of roles can be spread out across the network topology.
2. Nodes in roles share similar social structure to one another
    - All nodes in the roles are expected to show similar connectivity patterns.
3. Nodes do not have to reveal their unique node identifier
    - This increases privacy as node identifiers are not needed. Intermediaries only need to reveal the identity of their role, and not their unique identifier.
4. Nodes from same role may be used interchangeably
    - This provides greater redundancy in message paths as there are more forwarding opportunities than restricting forwarding to specific nodes.
5. We can route through the role connectivity graph
    - This increases scalability as the role connectivity graph is smaller than the full network graph, and thus takes up less memory.

We take advantage of these properties in the forwarding algorithm, SRR, by restricting the flood of messages based upon role membership.

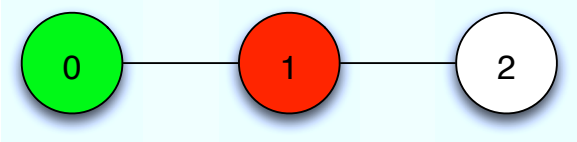


Figure 5.3: An example role connectivity graph (RCG). Nodes in role 0 only connect to nodes in role 1, similarly nodes in role 2 only connect to nodes in role 1. Nodes in role 1 can be seen as central, as they enable nodes in role 0 and 2 to communicate with each other.

5.1.2 Determining roles for forwarding

To categorise nodes in an opportunistic network into roles, we employ the social science technique of regular equivalence [144]. White et al. consider that "if the graphs for two types of social relation are identical, i.e. if they consist of exactly the same ordered pairs of nodes, the two will be treated as a single type." This approach has been considered to be too strict a restriction [123], because it requires that all nodes in the same role have to be tied to the same specific nodes. Instead, regular equivalence was introduced [144] and can be thought of as "a partition of nodes into classes such that nodes of the same class are surrounded by the same classes of nodes" [17]. In the aforementioned business example, all nodes in the manager role connect to at least one node in the assistant manager role, who in turn connect to at least one node in the non-managerial worker role. These roles can then be used to drive routing decisions. Consider three roles $(0, 1, 2)$, connected in the following manner: $\{0, 1\}$,$\{1, 2\}$ (Figure 5.3). If node a in role 0 intends to pass a message destined for node d in role 2 to intermediate node b in role 1, but instead meets node c from role 1, it may be beneficial for a to pass the message to c, as a and b may never meet. Note that nodes b and c do not need to ever have contact.

5.1.3 Partitioning the Network

Structural equivalence measures the way in which a node relates to the specific nodes in its neighbourhood, where the size of neighbourhood depends on the specific structural equivalence algorithm considered.

Regular equivalence, however, measures the way in which nodes relate to other nodes

across the whole network graph. Regular equivalence is, therefore, more useful for opportunistic routing than structural equivalence. Using regular equivalence we can partition nodes into roles, where members of roles can be assumed to have similar properties. Roles can be considered useful for opportunistic routing for the following reasons:

- Nodes can be in the same role as nodes they have not encountered, which provides greater resilience of groups in comparison to grouping via clustering
- Nodes in roles are similar in terms of social structure, which we could use to make assumptions on specific node behaviour based on the expectation of similar behaviour to other members of the node's role
- We can abstract the node identifiers away, which allows for greater privacy because node identifiers not needed
- Nodes from same role may be used interchangeably, which provides redundancy. This property provides forwarding opportunities based on roles, and is the property we exploit
- We can route through the role connectivity graph. This provides scalability, because nodes only need to know the connections between roles rather than nodes to be able to make assumptions about other roles behaviour.

As highlighted in Chapter 2, there exist several different mechanisms for calculating the regular equivalence classes (roles), but many of these mechanisms are difficult to automate. For instance, Blockmodelling [145] requires human involvement to determine the most appropriate number of classes, while CATREGE [17] works only on directed networks, which precludes the use of bi-directional relationship data such as a Facebook SRSN. We employ the Kanellakis-Smolka algorithm [78] as this provides a deterministic approach that works on directed and bi-directed networks, making it appropriate for all SRSN graphs.

The Kanellakis-Smolka algorithm is a fast implementation of Paige and Tarjan's algorithm for the Relational Coarsest Partition Problem (RCPP) [113]. RCPP is analogous to regular equivalence. We can see an example of this approach in Figure 5.4.

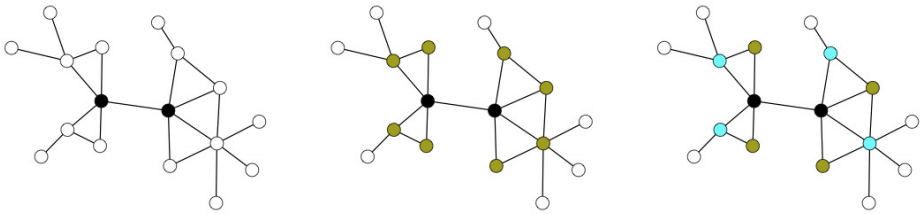


Figure 5.4: Here we see an input partition of two groups being iteratively broken down to form four roles, using the Kanellakis-Smolka regular equivalence mechanism. [140, p 240] First, we see the input partition, where we have determined that the two black nodes form a role and the other nodes form a role. Notice that under the definition of a role, all members of the role must have ties to the same roles as other members of the role. Thus, in the second figure, we split the white role so those nodes that have ties to the members of the black role form a new role. Third, the members of newer yellow role do not have consistent role ties: some members of the role connect to members of the white role as well as members of the black role. We therefore create a new blue role, consisting of those members of the yellow role with ties to members of the yellow, white and black roles. At this point, all members of all roles have ties to the same roles as the other members of their role, and our role partitioning is now complete.

To compute the social roles using the Kanellakis-Smolka algorithm, the network must be initially partitioned according to some selection criteria, for example:

- popularity – the number of unique nodes encountered
- betweenness centrality – the extent to which a node lies on shortest paths between other nodes
- information flow – the direction of message passing between nodes
- sociability – the proportion of time a node spends in the company of other nodes

In comparison to popularity or number of connections, betweenness provides a measure of the potential control over the passing of information between nodes, because of how it correlates to the number of shortest paths on which a node lies. We therefore use it to partition the input graph.

In our analysis we use betweenness centrality to make our initial partition (as suggested in [17]), place nodes with matching betweenness in the same class, and use the

Kanellakis-Smolka algorithm to compute the role assignments. This results in a graph of connections between roles, or a Role Connectivity Graph (RCG).

Strict regular equivalence results in many roles having size 1, and a number of roles similar to the number of nodes in the input graph. Roles of such a small size are impractical for forwarding, because if the sole node in a role is unavailable, there is no alternative node available for forwarding. Thus, we relax this constraint and enforce a minimum role size of 2 nodes.

5.1.4 Role divisions

As the performance of an opportunistic network may vary depending on the connectivity patterns of the nodes, we use four different real-world wireless data traces in our analysis:

1. The "SASSY" connectivity data [13] (as discussed in Chapter 4.3).
2. The Reality Mining trace from MIT [40] (as discussed in Chapter 4.3).
3. The HOPE dataset [1] (as discussed in Chapter 4.3).
4. The student schedule trace from NUS [132]. This trace consists of a timetable of 22,341 students' lectures (4,885 sessions, with an average number of 40.25 students in a session). The timetable is presented as contiguous hours without breaks overnight. Students in the same lecture are assumed to be in contact with one another. To improve realism we assume a 6 day work-week with each working day lasting 11.5 hours (work day plus commute time). As this is a dense dataset, to reduce simulation complexity we use the method described in [95] to reduce the number of nodes, whilst preserving the degree distribution. As the trace is a timetable, we can view it as a SRSN, as this is a list of declared connections between individuals. We can also use this trace as an encounter list, as it is a record of when individuals should meet one another. To create a more realistic scenario, we randomly remove 34% of nodes each day from the encounter trace, as we assume that these students did not attend classes [131]. The DSN encounters are condensed as follows: bundles of messages are assumed to be passed on within 30 seconds, so any encounters less than

Table 5.1: Trace properties: all traces

Trace	SRSN	DSN
SASSY	Facebook Friends	ZigBee Encounters
HOPE	Topic Interest	RFID Co-Location
RM	Phone Contact List	Bluetooth Encounters
NUS	Student Timetable	Reduced Student Timetable

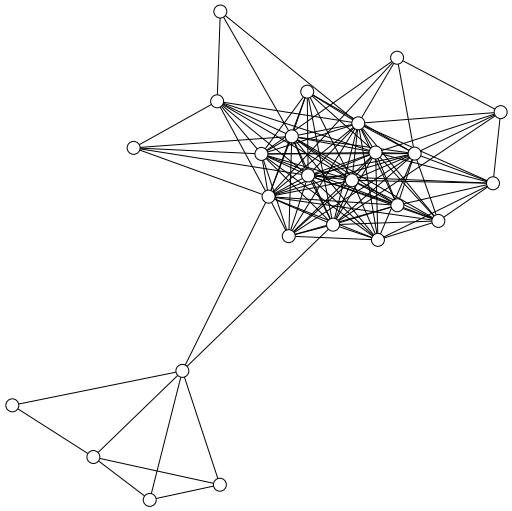
this period of time were not included in the simulations.

We can see the details of the input data used to create the SRSN and DSNs in table 5.1. We believe these traces capture a wide variety of possible network scenarios. We can see in Table 5.2 that the number of nodes varies in each trace, from 25 to 500, with a similar variation in the number of connections between the nodes in each trace. The SRSNs for these traces can be seen in Figures 5.5a–5.5d. The SASSY SRSN has one large component, the RM SRSN has two large components, four pairs of nodes and 23 disconnected nodes. The NUS trace has two components, the nodes of which are highly connected. The HOPE trace has one large connected component.

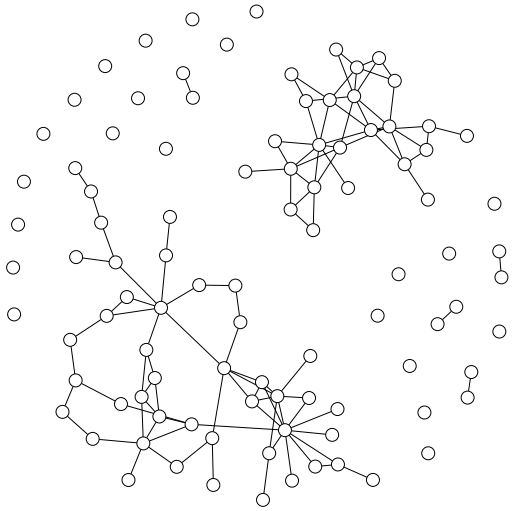
5.1.5 Role Analysis

We now describe the role assignments of nodes in the four wireless traces. Figures 5.6a–5.6d show the RCGs for the SASSY, RM, NUS and HOPE traces respectively. These graphs show the connection patterns between members of the roles. Each vertex represents one role, and may have two or more members. Summary statistics of each input trace SRSN graph and output RCG can be seen in Table 5.2.

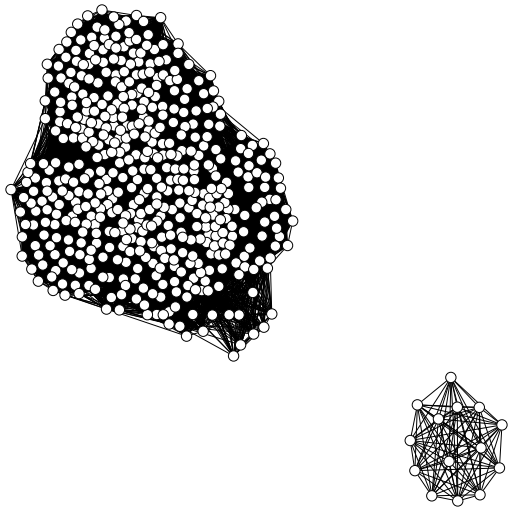
The graph density is higher in each RCG, meaning, that on average, each node needs to consider more connections using the RCG than a purely SRSN-based protocol. In all cases the number of vertices has been reduced from the input partition, meaning that storing the RCG requires less memory than storing the whole SRSN graph.



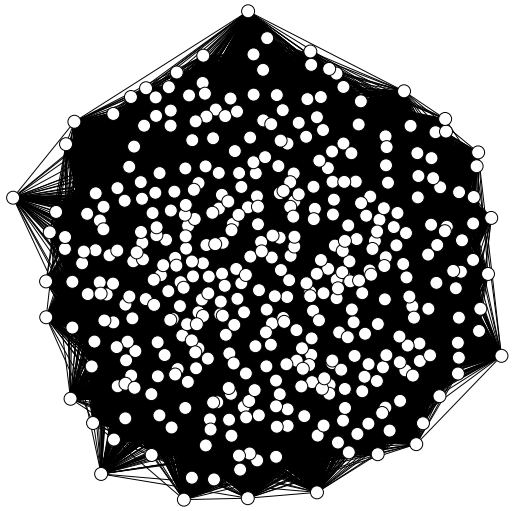
(a) SASSY trace. Two groups of nodes are linked by a single node.



(b) Reality Mining trace. There are two disconnected groups of nodes, four groups of outlying pairs, and 22 disconnected nodes.



(c) NUS trace. There are two distinct groups of highly connected nodes.



(d) HOPE trace. Here we we see a large connected group of nodes.

Figure 5.5: The self-reported social networks for our four chosen datasets.

Table 5.2: Dataset graph statistics

	Trace							
	SASSY		Reality Mining		NUS		HOPE	
Property	SRSN	RCG	SRSN	RCG	SRSN	RCG	SRSN	RCG
Number of Vertices	25	5	97	16	500	26	414	27
Number of Edges	127	7	107	23	29743	198	67836	329
Clustering Coefficient	0.748	0	0.255	0.246	0.605	0.808	0.834	0.885
Graph Density	0.406	0.560	0.023	0.180	0.238	0.586	0.792	0.903
Graph Components	1	1	28	3	2	1	1	1

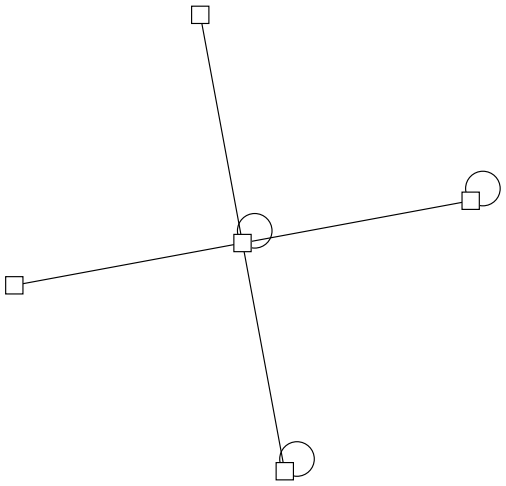
The SASSY and RM RCGs have central hub roles that connect the other roles together. We believe that nodes in this group are likely to be overloaded by existing routing protocols.

Central roles can be seen in both the SASSY and RM RCGs. Notice that in the RM SRSN several nodes have no ties to other nodes, and using a forwarding protocol based purely on the SRSN data might lead to the assumption that these nodes make no contacts during the trace. In the RCG graph nodes which have not ties in the SRSN are now members of a role. We also see that the two large disconnected groups are now connected together. This shows that role analysis can find similarities between nodes that are not in the same components.

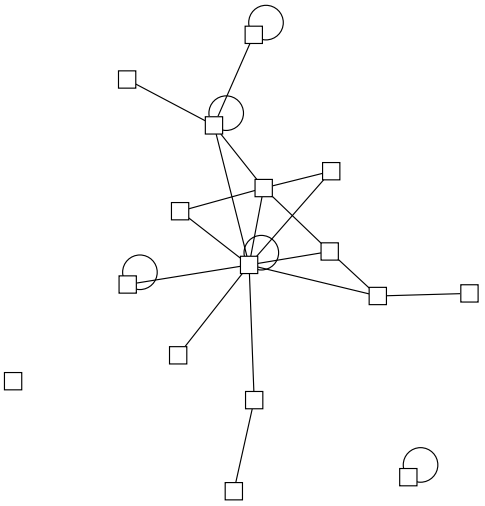
The RCG for the NUS trace data (Figure 5.6c) shows a group of highly connected nodes, with six outlier nodes, each with four or fewer connections. The HOPE trace's RCG is similar to its SRSN connectivity graph (Figure 5.5d), and shows a large group of roles.

5.1.6 Social Role Routing

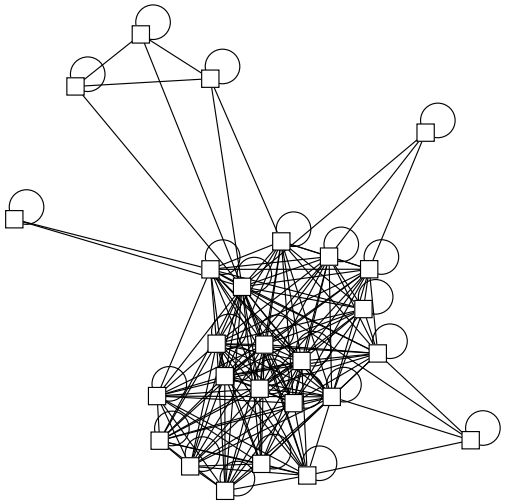
We now describe a routing protocol for opportunistic networks that uses social roles, Social Role Routing (SRR).



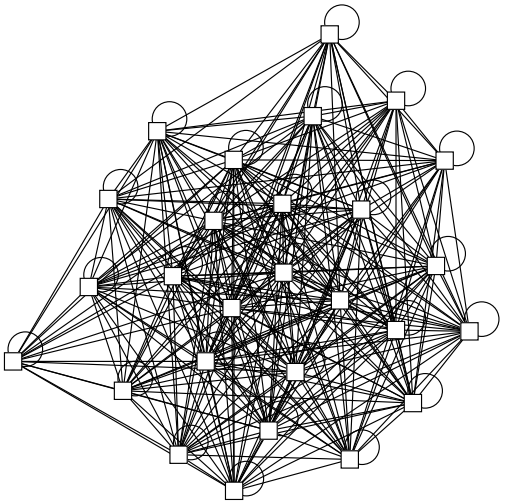
(a) SASSY trace. A central role connects two clique roles, with two other roles whose nodes only connect to nodes in the central role.



(b) Reality Mining trace. There is a highly important central role. All the outlying nodes (Figure 5.5b) have been grouped together.



(c) NUS trace. There is a large group of highly connected central roles, with a small number of roles outside this group.



(d) Hope trace. We see a large connected group of roles.

Figure 5.6: Role Connectivity Graphs for the four different datasets, showing the interconnections between the node roles in the Self-Reported Social Network. Distances are meaningless, squares indicate roles, and loops indicate members of the role connect to one another.

As reducing message duplication can prolong battery lifetime we only allow forwarding of messages to intermediate nodes that are in the same role, or in a role adjacent to, the destination's role in the RCG. As they are likely to see nodes in the destination's role, and therefore by extension, the destination, nodes in or adjacent to the destination's role are appropriate candidates for intermediary nodes.

Before the network starts up, we analyse the SRSN of the participating nodes using the method described in Section 5.1.2. Each node stores a copy of the resulting RCG, allowing them to compute the geodesic distance between roles. Each node has a unique identifier (ID) and stores the identifier of the role to which it belongs (RoleID).

We can see the pseudo-code for SRR in Algorithm 1. When nodes encounter one another, each provides its ID, RoleID and identifiers of each of the messages that it is carrying. For each message in its message buffer not present in the other node's message buffer, the node checks the geodesic distance of the encountered node's role from the destination node's role. If this distance is less than or equal to 1 (i.e., it is in the same role or in a role adjacent to the destination's role), the node duplicates the message and passes it on to the encountered node.

Nodes duplicate each message, rather than passing on the only copy, to increase the number of paths that the message may take. We believe that the forwarding constraint of passing to only nodes in the destination's role or the adjacent role, will prevent nodes from flooding the network.

Algorithm 1 SRR forwarding algorithm

1: $r \leftarrow$*Maximum Role Distance Allowed*

function EncounterNode(B):

1: **for all** *message* in message_buffer **do**
2: $currdist \leftarrow$ getRoleDistance(*message.destination.role* , *B.role*)
3: **if** $currdist \leq r$ **then**
4: giveMessage(*message*,*B*)

5.2 Simulation Setup

To evaluate SRR, we perform trace-driven simulation using the four traces described in Section 5.1.4, with five routing protocols.

5.2.1 Routing protocols

We simulate established routing protocols where the source code is available and the protocol can be validated from the paper describing the protocol. For example, the PeopleRank [108] protocol would have been appropriate to simulate, but the source/pseudo code is not available.

We evaluate the following protocols:

- Epidemic [142]: Nodes forward messages to any node that they encounter that do not already have a copy of the message.
- SimbetTS [35]: This is an example of a well-known existing history-based opportunistic routing protocol. Nodes forward messages depending on a utility value computed from node similarity, betweenness, and tie-strength. A small number of message duplications are allowed.
- Simple Social Network Routing (SSNR): The simple social network routing protocol from Chapter 4. Nodes embed their SRSN into each message they create. Copies of the message are only forwarded to nodes in the SRSN.
- Social Role Routing (SRR): The SRSN is used to determine roles, and routing is performed as described in Section 5.1.6. Nodes forward only to nodes in the destinations role, or in a role adjacent to the destination.
- Social Role Routing SimbetTS Hybrid(HySimbR): Many routing protocols (such as SimbetTS) use a warm-up period in which to gather enough data to make accurate routing decisions. In this protocol we switch from the SRR protocol to SimbetTS after 15% of the encounters. This allows us to study the performance of the system using SRR to bootstrap the network before switching to a history-based protocol.

5.2.2 Routing Evaluation

To evaluate SRR against existing protocols, we use trace-driven simulation of Nokia N95 smartphones running a messaging application.

We investigate two different simulation scenarios. In our first scenario we simulate opportunistic routing using a segment of each of the four encounter traces. In our second scenario we examine the behaviour of the routing protocols as they warmup.

For the SASSY and RM traces we use the initial 30 days of the trace. We simulate seven days in the shorter NUS trace, and one day of the HOPE trace. Table 5.3 describes the simulation parameters, which are in line with existing work on opportunistic networks, e.g., [57].

In previous work, researchers have used the initial 15% of encounters as a warm-up period [35]. In our second scenario we analyse only the first 15% of encounters (roughly five days for a 30-day trace). This is to examine how routing protocols perform in the early stages of the network lifetime. We hypothesise that history-based protocols will perform poorly during this period (note that simulating the HySimbR protocol in this scenario is redundant and therefore not discussed).

5.2.3 Messaging and Battery Model

Each node is assumed to be a Nokia N95 smartphone, with the message buffer and battery parameters given in Table 5.3. We only consider encounters over 30 seconds in length, assuming that shorter encounters are insufficient for exchanging data.

The message senders and sources are chosen from a randomly-generated exponential distribution, to model real-world messaging patterns.

To determine an energy model for our nodes, we enabled the Bluetooth radio on a Nokia N95 smartphone with a 1200 mAh battery and measured energy depletion using the Nokia Energy Profiler v1.2. Our simulated nodes' batteries deplete according to this observed behaviour (1.9×10^{-6}mAh). When a node sends a message, it decrements its

Table 5.3: Simulation parameters

Parameter	Value
TTL of messages	10 days / 1 day / 2 hours
Message frequency	1 per node per day
Simulation length (days)	30 / 7 / 1
Simulation runs	30 per protocol
Message size (MB)	1
Buffer size (MB)	2000
Loss per second (mAh)	1.9×10^{-6}
Time to send bundle (s)	34
Max energy (mAh)	1200
Energy per send (mAh)	0.4
Charge time (h)	8

battery level by 0.4mAh. When a node runs out of battery it charges for 8 hours, and is not present on the network during this time.

5.2.4 Metrics

To compare the routing protocols, we analyse the following commonly-used metrics. These show the performance of the protocol across the nodes in the network:

- Delivery ratio: The ratio of the number of delivered destinations over the total number of destinations. A protocol that delivers a large proportion of messages sent can be seen to have a high performance [19, 35, 65, 72, 142].
- Delivery cost: The average number of medium accesses required for delivering a data item to a destination [72]. A protocol that has a low cost can be seen to preserve the battery of the nodes more than one with a higher delivery cost.
- Forwarding distribution: The standard deviation of percentage of total forwards performed by each node. A protocol in which more nodes share the burden of forward-

ing messages may have an extended network lifetime.

5.3 Results

In this section we present the results of the trace-driven simulations performed to evaluate our opportunistic routing protocol. First, we discuss the full-length scenario, and then the warmup period scenario. Finally we discuss the analysis of the distribution of forwarding amongst the nodes.

5.3.1 Full-length scenario

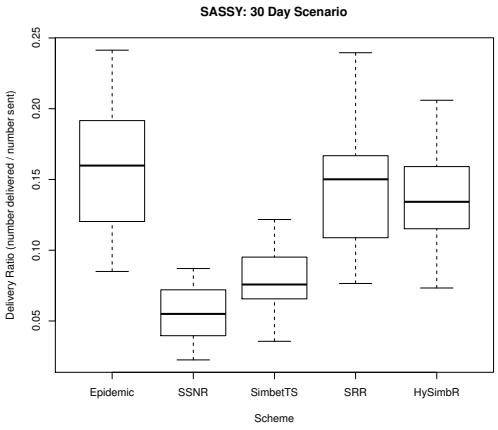
This scenario is indicative of performance of the protocols up to and achieving expected operating performance. No one protocol performs well across all of the traces.

Figures 5.7a–5.7d show that the role-based protocols SRR and HySimbR perform equivalently to, or better than, SimbetTS. This is because using SRR allows for more message duplications than SimbetTS and SSNR.

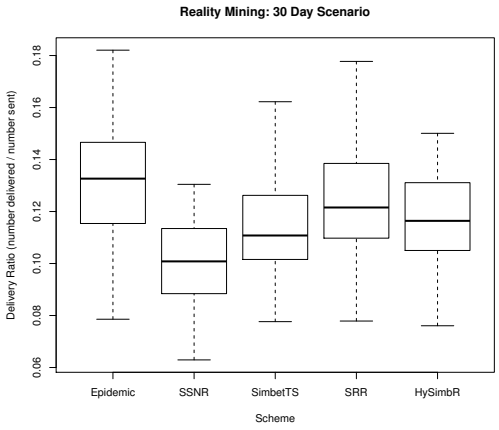
When we look at the cost of forwarding (Figures 5.8a–5.8d), we see that the role-based protocols do have a higher cost than SimbetTS due to extra message duplications. Despite the higher cost, however, SRR and HySimbR maintain a delivery ratio similar to Epidemic routing, while having a lower cost. The extra forwarding cost that SRR and HySimbR incur over SimbetTS is not, however, enough to reduce the delivery ratio. Using SRR or HySimbR would therefore provide better performance.

5.3.2 Bootstrap scenario

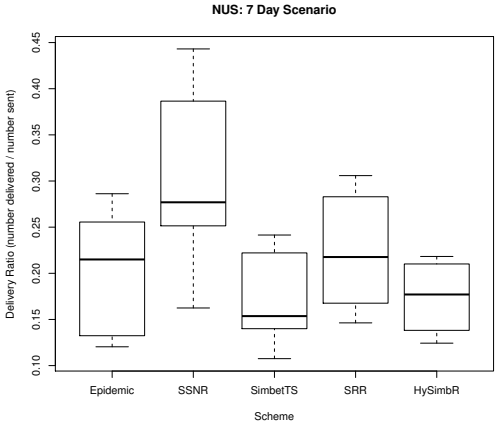
This scenario captures the performance of protocols during the protocol warmup phase. In contrast to the full-length scenario where no one protocol consistently outperforms the others, SRR always does better than SimbetTS in the bootstrap scenario. Figures 5.9a–5.9d show that SRR outperforms SimbetTS in all four traces, although the difference between



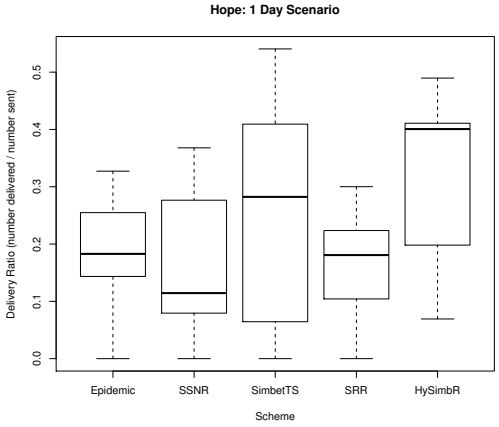
(a) SRR and HySimbR outperform SimbetTS. There is no significant difference between Epidemic and either SRR or HySimbR.



(b) SRR outperforms SimbetTS, and is not significantly different to Epidemic. The difference between SimbetTS and both SRR and HySimbR is not statistically significant.

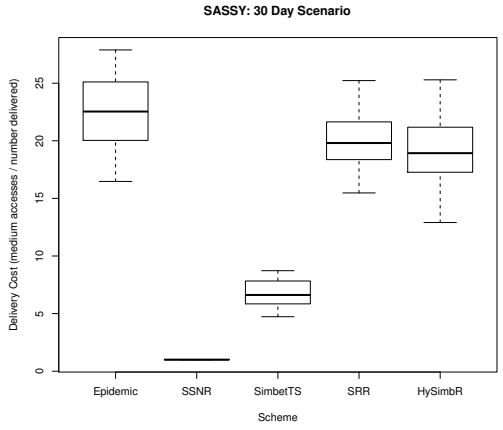


(c) SimbetTS performs slightly worse than HySimbR, and SRR marginally outperforms Epidemic. SSNR performs the best because it uses a lower number of forwarding paths than SRR, but more than SimbetTS.

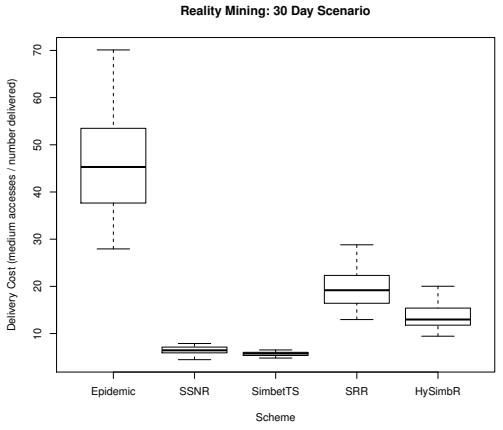


(d) HySimbR performs the best, followed by SimbetTS. Epidemic, SSNR, and SRR are equivalent.

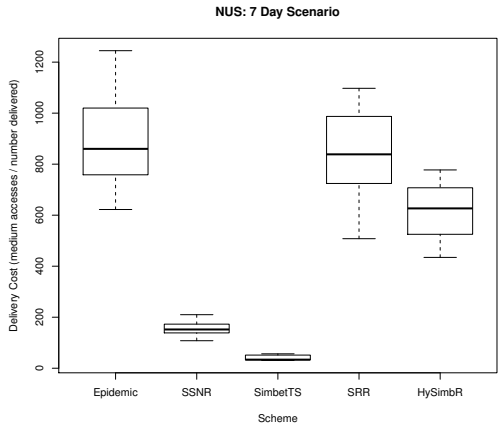
Figure 5.7: Delivery ratios for the full-length simulations. We see that SRR and HySimbR perform as well as, or better than, SimbetTS in all scenarios.



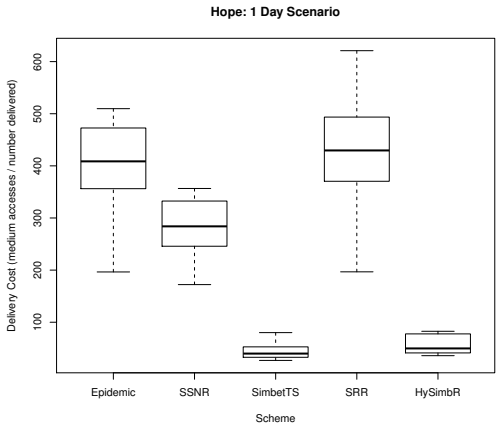
(a) SimbetTS has a lower cost than Epidemic and the role-based schemes. SRR and HySimbR, however, have a lower cost than Epidemic routing.



(b) Epidemic routing has the highest cost. There is no statistically significant difference between SSNR and SimbetTS.



(c) We see that HySimbR has a lower cost than SRR and Epidemic, and SimbetTS performs better than all other schemes.



(d) Here we see that SRR and Epidemic have a similar cost, as do SimbetTS and HySimbR.

Figure 5.8: Delivery cost for the full-length simulations.

the other protocols in the NUS trace is not statistically significant.

Figures 5.10a–5.10d show that SRR has a lower cost than Epidemic routing, but a higher cost than SimbetTS. We believe, however, that SRR is preferable in this scenario as it is computationally simpler than SimbetTS, offers a delivery ratio similar to, or better than, SimbetTS, and similar to Epidemic, with a delivery cost similar to, or lower than, Epidemic routing. As we saw in the full length scenario, it is beneficial to switch to another protocol after a certain period of time, before the increased cost due to the number of duplications impacts delivery ratios.

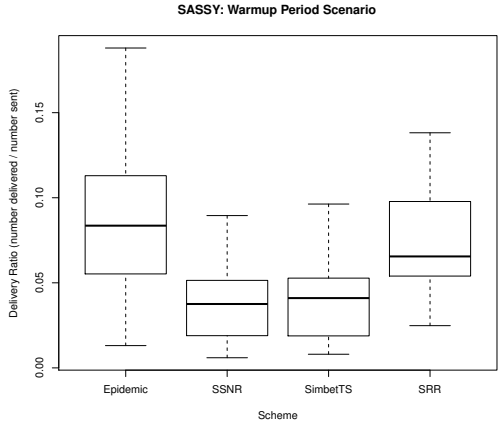
SimbetTS has a lower cost than Epidemic and SRR, which can especially be observed in the NUS trace (Figure 5.10c). Despite their high cost, Epidemic and SRR are not overloading the devices battery or storage to a level that adversely affects routing performance, and we observe high delivery ratios. This shows that minimising delivery cost at the expense of increased message duplications need not necessarily be a priority of routing protocols.

Despite the differences between the traces, SRSNs and RCGs in terms of density, clustering coefficient and number of vertices, we see that SRR is the best of the evaluated protocols to use to bootstrap the network.

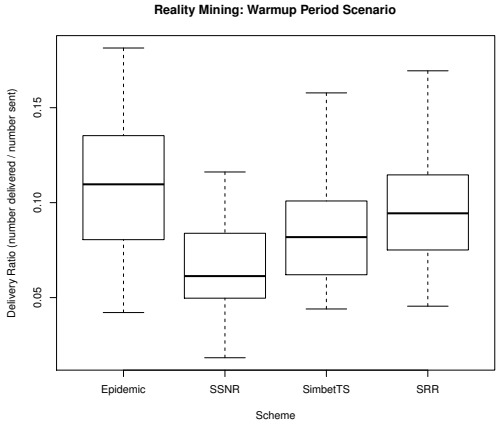
5.3.3 Distribution of Forwarding

Protocols that require a small subset of nodes to do a large amount of the forwarding may incur costs to those nodes that cause them to quickly run out of battery. If nodes do run out of battery then the network diameter may increase, resulting in poorer network performance.

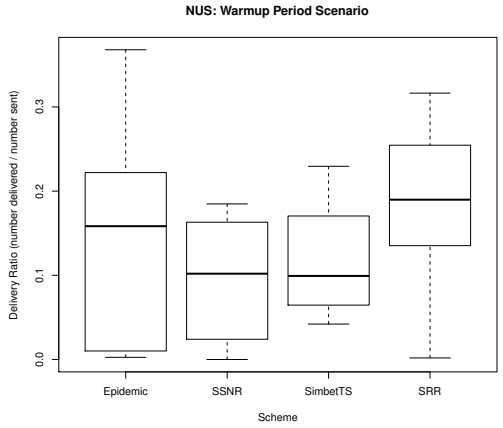
SimbetTS does badly in the dense traces, NUS and HOPE. We see in Figures 5.13a and 5.13b that SimbetTS has a worse cost distribution than SRR in the NUS trace, and the worst cost distribution in the HOPE full length scenario Figure 5.14b. In both NUS and HOPE traces, HySimbR always outperforms SimbetTS. This is because SimbetTS requires a specific group of nodes to do forwarding, whereas SRR and HySimbR share the forwarding responsibility amongst nodes.



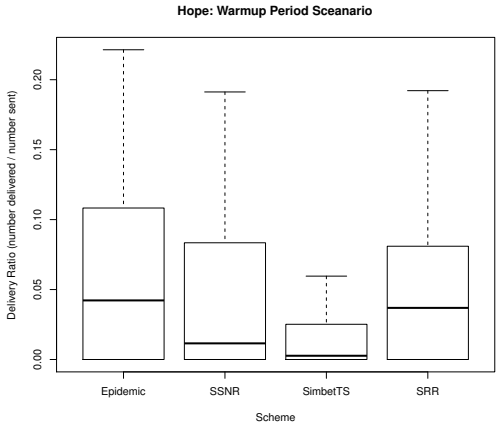
(a) SRR performs better than SimbetTS, and has no significant difference in performance to Epidemic routing. SSNR has no significant difference in performance to SimbetTS.



(b) Again SRR has no significant difference to Epidemic or SimbetTS.

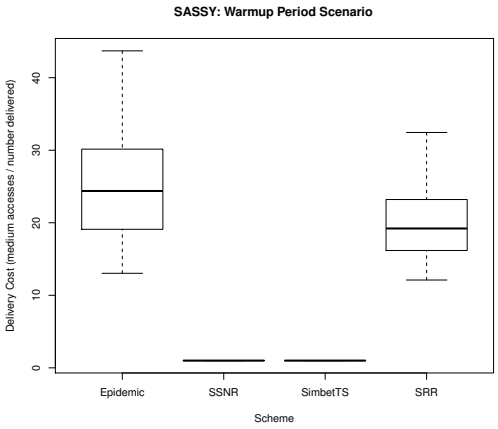


(c) SRR performs slightly better than the other schemes.

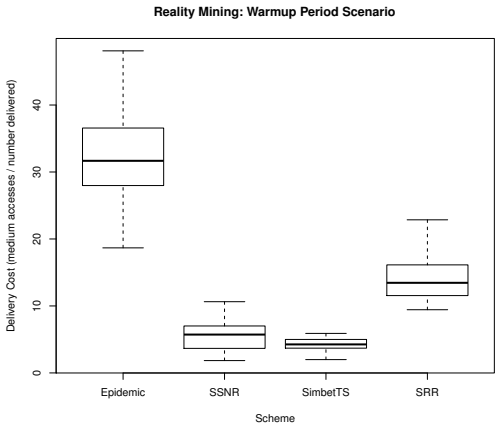


(d) SimbetTS performs slightly worse than the other schemes. There is, however, no statistically significant difference between the schemes.

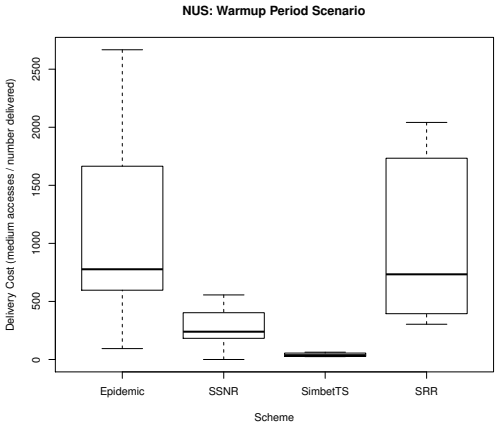
Figure 5.9: Delivery ratios for the four traces during the bootstrap scenario.



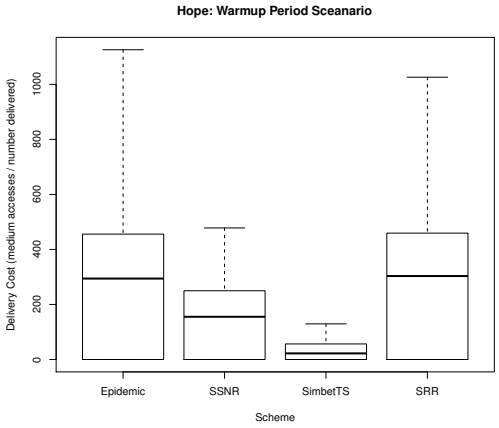
(a) SRR has a lower delivery cost than Epidemic routing. The difference is not, however, statistically significant.



(b) SRR has a lower cost than the Epidemic scheme. There is no statistically significant difference between SSNR and HySimbR.

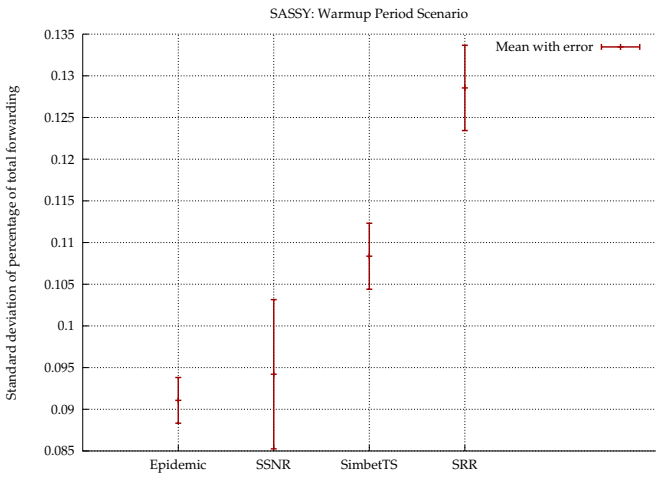


(c) There is no significant difference between the Epidemic, and SRR. SimbetTS performs the best out of all the traces.

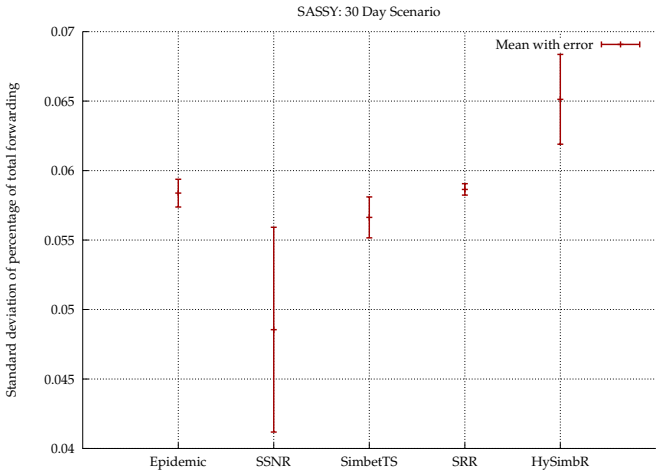


(d) SimbetTS has the lowest cost. There is no statistically significant difference between the rest of the protocols.

Figure 5.10: Delivery cost for the four traces during the bootstrap scenario.

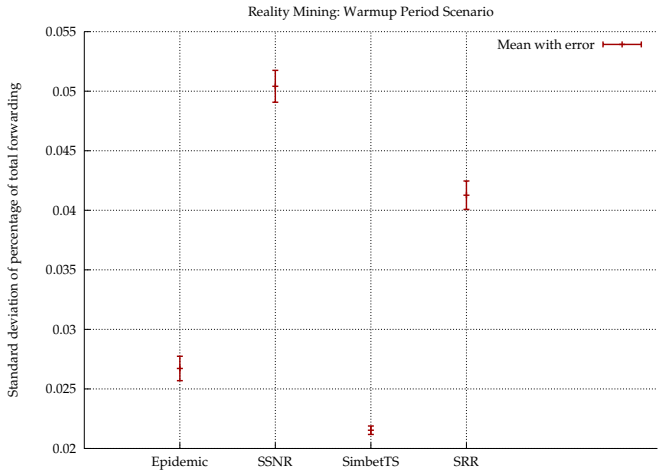


(a) SRR performs the worst. SSNR and Epidemic are equivalent.

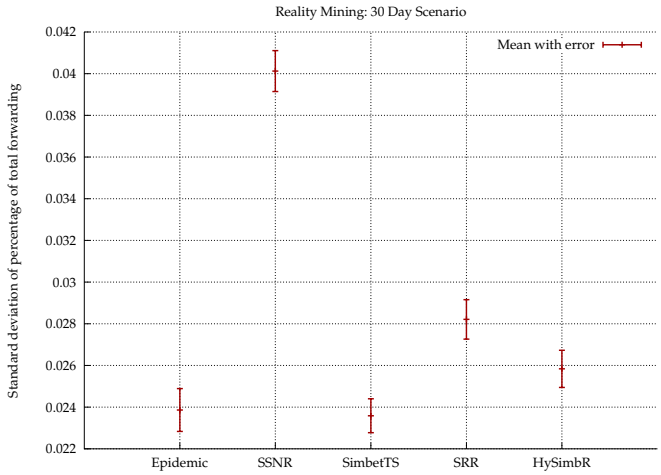


(b) Epidemic, SRR, and SimbetTS are all equivalent.

Figure 5.11: Standard deviation of the percentage of forwarding performed by nodes in the SASSY trace.

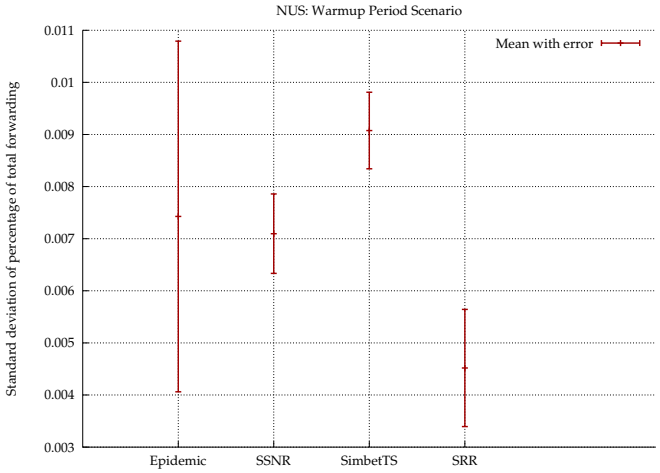


(a) SimbetTS performs the best, followed by Epidemic, followed by SRR, with SSNR coming in last. All differences are statistically significant.

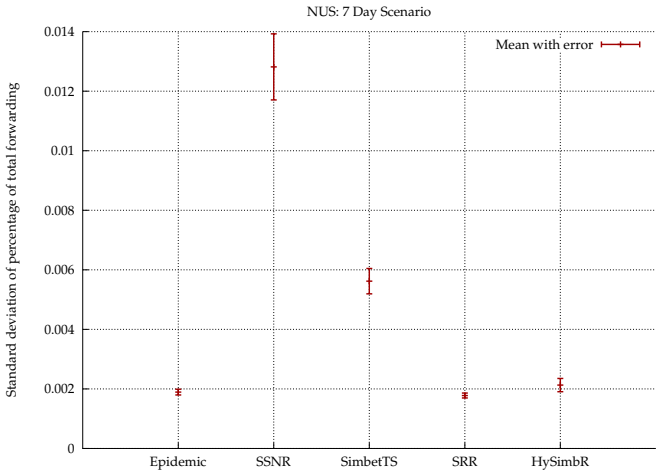


(b) Epidemic and SimbetTS are equivalent. HySimbR performs better than SRR, and SSNR.

Figure 5.12: Standard deviation of the percentage of forwarding performed by nodes in the RM trace.

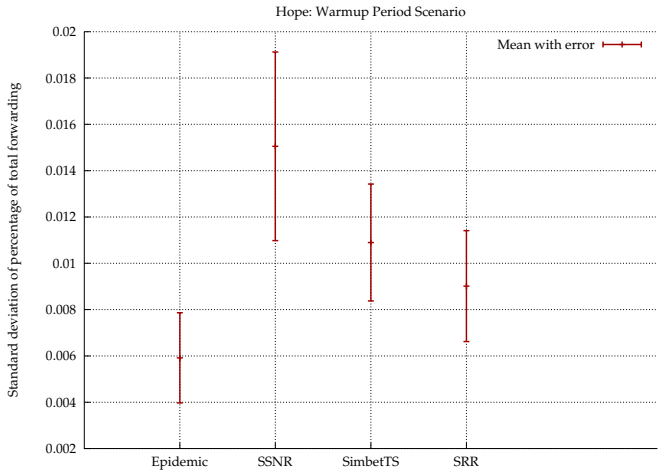


(a) There is a large error value for the Epidemic routing protocol. SRR has a lower mean standard deviation than SSNR and HySimbR.

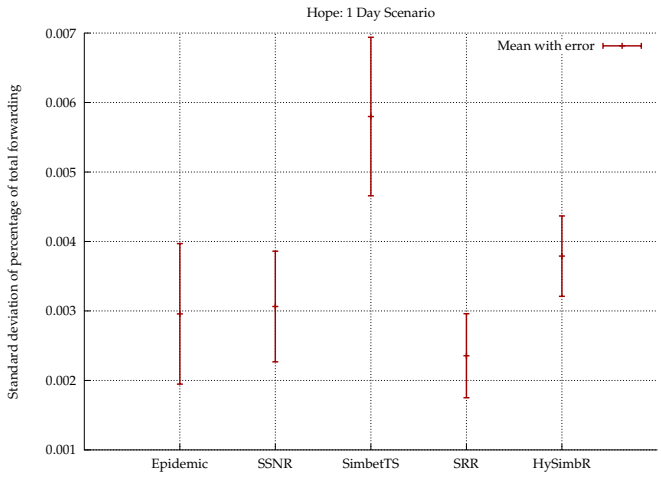


(b) SSNR performs the worst, followed by SimbetTS. There is no significant difference between Epidemic and SRR, nor between Epidemic and HySimbR.

Figure 5.13: Standard deviation of the percentage of forwarding performed by nodes in the NUS trace.



(a) SSNR performs the worst. SRR is no different to SimbetTS or Epidemic. Epidemic performs better than SimbetTS.



(b) SimbetTS has the highest standard deviation of forwarding performed. Epidemic and SSNR are equivalent to HySimbR and SRR. SRR performs better than HySimbR.

Figure 5.14: Standard deviation of the percentage of forwarding performed by nodes in the HOPE trace.

SimbetTS is similar to SRR in the SASSY trace during the full length scenario, and better than the HySimbR. SimbetTS does very well in the RM bootstrap scenario, outperforming all the other protocols. SimbetTS's better distribution of forwarding in the RM trace (and good performance in the SASSY trace), does not translate into better delivery performance, as discussed in Section 5.3.2 and 5.3.1, where it performs equivalently to SRR and HySimbR.

Epidemic routing has a low deviation of forwarding in most of the traces. We know from the cost analysis that Epidemic has a high delivery cost, however, and that this high cost impacts delivery ratios. We should therefore consider the delivery cost when deciding whether a more even forwarding distribution is important.

SRR works well in the dense traces, performing the best, or equivalently best, in NUS and HOPE. These traces have a high delivery cost (Figures 5.10c and 5.10d), so maintaining an even distribution of forwarding is important for preserving forwarding paths.

There is no best protocol in terms of distribution of routing. None of the protocols have an consistently even distribution of forwarding across all traces or scenarios. We do see that SRR maintains an even forwarding distribution in the dense traces NUS and HOPE, distributing forwarding in traces where a lot of forwarding is required. Even though SRR does not have the most even forwarding distribution of the assessed protocols in the sparser traces (RM and SASSY), where less forwarding is required, it still maintains a high delivery ratio. The uneven distribution in these traces is caused by the lack of alternatives for forwarding when the trace is sparse.

5.4 Conclusions

We have presented a novel opportunistic routing protocol, Social Role Routing, that uses self-reported social networks and applies the social network analysis technique of role analysis to select nodes to act as message relays. This allows the protocol to route effectively when expected nodes are not present, and to provide good performance while encounter-history-based protocols are still warming up.

We compared our protocol to four existing protocols, and found that SRR performs as well as existing protocols in most scenarios. Moreover, SRR always performs better than existing protocols when used to bootstrap an opportunistic network, with a delivery ratio similar to Epidemic routing, at a much lower cost. Over the full trace length it may perform as well as SimbetTS, but at a higher cost, so we proposed and evaluated a hybrid protocol which provides the best of both approaches.

We have seen that minimising delivery cost is not necessarily an indication that a routing protocol will perform better than a protocol that does not. SRR performs well despite having a high cost, as it aims to avoid adversely overloading nodes that perform a large bulk of the forwarding. We have seen in dense traces where the cost of forwarding is high, that SRR shares the forwarding cost amongst the nodes more evenly than (or as evenly as) other protocols, thus maintaining a high delivery ratio.

We have evaluated the protocols' performance using four traces, each with differing structure and connectivity patterns. No one protocol consistently performs best across all of these traces. Which trace, if any, is indicative of an opportunistic network in the general population, is an open question. Even then, it appears that a system designer's choice of routing protocol may depend on the nature of the users of the proposed system.

In this chapter we have demonstrated the following:

- we can create routing protocols that work without encounter histories, and instead use social roles to route effectively
- we can use SRSNs to create the role information, thereby creating a role protocol SRR, that can effectively bootstrap an opportunistic network

In this chapter we have discussed protocols for opportunistic network routing. Nodes must, however, adhere to the routing protocol for the network to work effectively. In the next chapter we therefore look at how to incentivise nodes to adhere to opportunistic routing protocols.

Chapter 6

Detecting Selfishness and Incentivising Cooperation in Opportunistic Networks

This chapter leads on from using social roles for forwarding, and focuses on how to combat selfishness in opportunistic networks using SRSNs. Many existing studies of opportunistic routing focus purely on network level metrics, for example: delivery delay, delivery cost and delivery ratio. Frequently, nodes may wish to avoid the costs associated with participation in an opportunistic network. To avoid the forwarding costs, nodes can cut down on the messages that they forward. If nodes forward only their own messages and not the messages of other nodes (against the behaviour expected of them by the routing protocol), we refer to this behaviour as selfish. When a node is behaving selfishly, it will not forward a message for another node at any time.

Many existing works do not consider the incentives for nodes to participate in the network. In this section we see how it is possible to detect such selfishness and how to incentivise nodes to cooperate according to the routing protocol. As outlined in Chapter 3 we need to develop an incentive protocol that works on multi-copy forwarding protocols in opportunistic networks.

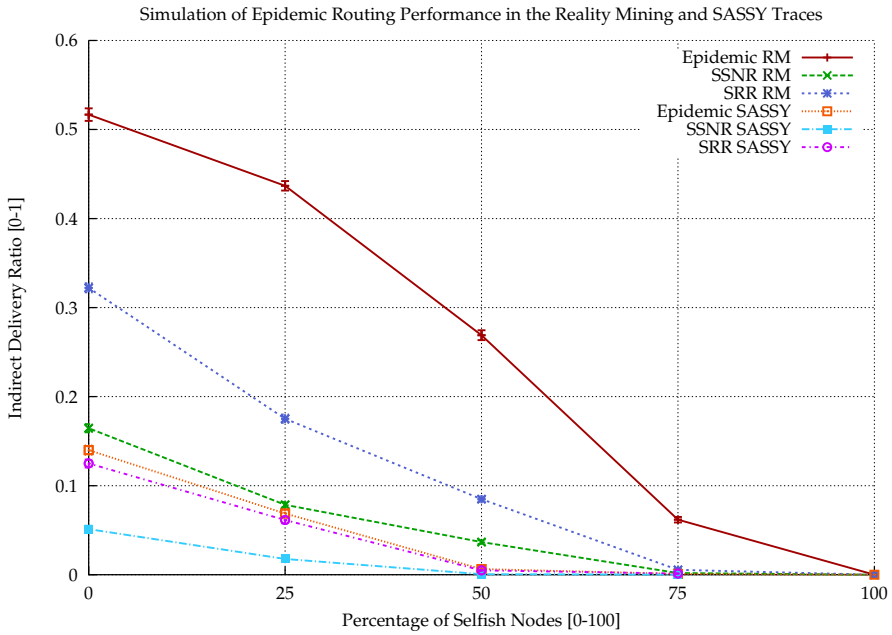


Figure 6.1: Delivery ratio opportunistic routing under differing levels of node selfishness.

This work was originally published with Dr Tristan Henderson in [12].

Opportunistic networking relies on cooperation between nodes, that is, the users participating in the network, to perform efficiently. Opportunistic routing protocols depend upon nodes forwarding messages for each other. The only delivery mechanism possible without forwarding/data muling would be for the creator of a message to encounter the message destination node and deliver the message directly. Cooperative forwarding, however, incurs a cost to the forwarding nodes, both in terms of energy (battery power) and storage (the space required to store forwarded messages). Both of these are constrained resources in mobile devices used in opportunistic networks.

Nodes may wish to avoid the costs associated with participation in an opportunistic network by not forwarding messages for other nodes. Figure 6.1 shows the results of an opportunistic network simulation where nodes act selfishly: the performance of the network, i.e., the number of messages delivered, decreases rapidly as the percentage of selfish nodes increases. If we can detect and discourage selfish behaviour it might be possible to achieve the same performance as the case where no nodes are selfish, even if all the nodes have a propensity for selfish behaviour.

How can we create incentives for nodes to cooperate? This chapter investigates how to use social network information to build an incentive mechanism for opportunistic networks.

We present an incentive mechanism for opportunistic networks, IRONMAN (Incentives and Reputation for Opportunistic Networks using sociAl Networks), that uses social-network information to bootstrap the detection and discourages selfishness.

We demonstrate IRONMAN's superior performance over two existing incentive mechanisms, and show how to improve existing mechanisms by using social-network information.

The contributions of this Chapter are:

- showing that existing incentive mechanisms are inappropriate for opportunistic networks.
- showing that social-network information can bootstrap a trust mechanism to discourage selfishness in opportunistic networks.

6.1 An Incentive Mechanism For Opportunistic Networks

Opportunistic networks exploit the interconnections of individuals as they go about their daily lives. In society we form ties and connections with people around us, be it work colleagues, friends or family. Our goal is to use a record of these social-network data from self-reported social networks (SRSNs) to bootstrap an incentive mechanism for opportunistic networks. SRSNs can be obtained through interview, or from an online social network (e.g., Facebook friends lists).

By viewing the members of the opportunistic network that are also in a node's SRSN as more trustworthy, we can exploit the implicit trust relationships provided by the users. Detecting selfish behaviour quickly reduces the amount of transmissions to (and due to) selfish nodes, and therefore the energy wasted. We must balance this, however, against being too cautious and presuming that all nodes are selfish. Most existing mechanisms

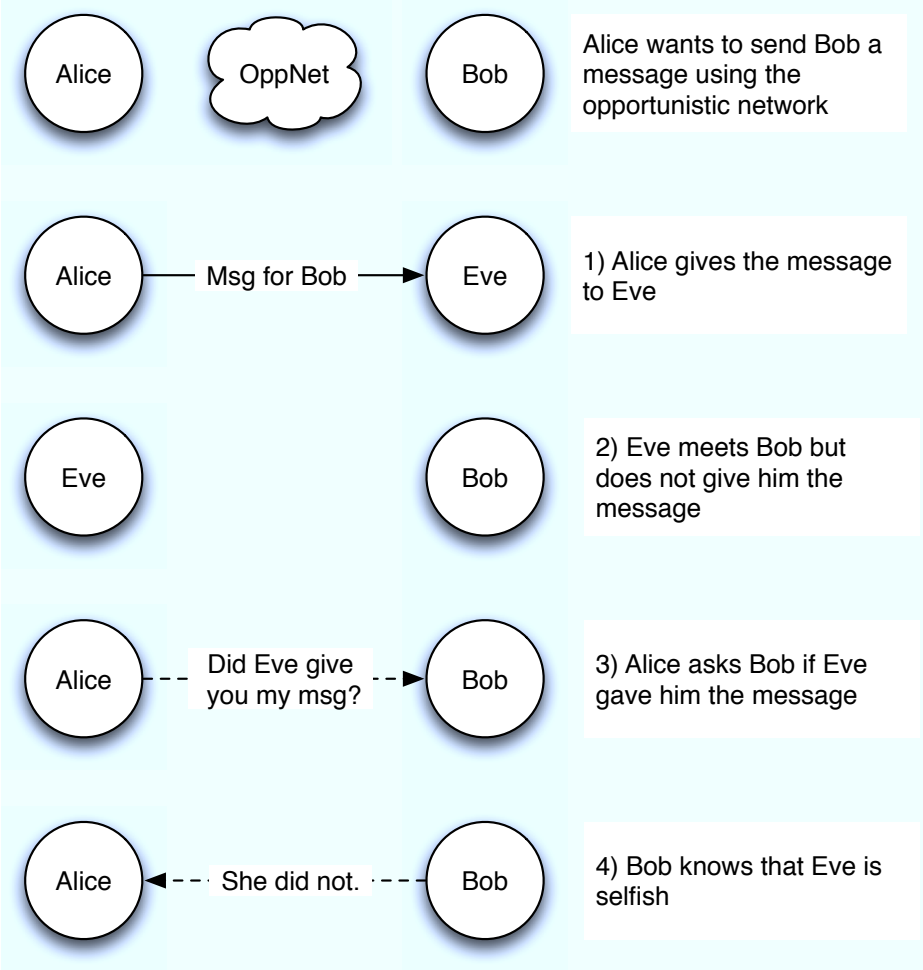


Figure 6.2: Nodes keep a history of their encounters and message exchanges. When nodes meet they exchange histories to detect selfishness.

are not bootstrapped to work from network start-up, we use SRSNs to bootstrap the incentive mechanism. We assume that individuals have implicit trust with members of their SRSN: therefore, when the network starts up, nodes assign higher trust values to nodes in their SRSN, and do not trust nodes not in their SRSN. The trust values may, however, be modified over time.

6.1.1 Detecting selfishness

We now present our IRONMAN mechanism. Consider the following scenario (Figure 6.2): Alice wishes to send a package to Bob. She meets Eve first, however, and gives Eve the

package, believing that Eve will meet Bob before Alice does. Eve then meets Bob, and because Eve is selfish, does not give Bob the package. Yet later, Alice meets Bob, and they discuss their encounters. Alice mentions to Bob that she gave Eve a package for Bob. Bob knows he met Eve, and therefore knows that Eve was being selfish by withholding the package. We extend this analogy to opportunistic networks.

If nodes can store a history of encounter times and messages exchanged, and exchange histories during encounters, we can detect selfishness and altruism as seen in Algorithm 2.

If a node detects another node as selfish, the detecting node decrements its rating for the selfish node by the *behaviour constant* x. Similarly, when nodes pass on a message for which they are not the source: the receiver marks them as altruistic, and their rating of the node receiving the message is incremented by x. Additive increase and decrease are used to reduce the effect of false positives. False positives can arise when a node pushes a message out of its buffers due to congestion (giving the appearance that it deliberately dropped the message).

Algorithm 2 IRONMAN selfishness detection

1: $x \leftarrow$*behaviour constant*

function EncounterNode(B):

1: history_tuples $\leftarrow [(exchange\_time, msg\_id, msg\_source, node\_seen)]$
2: exchange forwarding history with B
3: **for all** *message_exchanges* in foreign_history **do**
4: **if** $exchange\_time >$ last encounter with B **then**
5: **if** $msg\_destination == my\_id$ **then**
6: **if** last encounter with $node\_seen >$ last encounter with B **then**
7: **if** *node_seen* did not give us *msg_id* **then**
8: $\text{Rating}_{node\_seen} \leftarrow \text{Rating}_{node\_seen} - x$

function ReceiveMessage(*other_node*, *msg_src*):

1: **if** $other\_node \neq msg\_src$ **then**
2: $\text{Rating}_{other\_node} \leftarrow \text{Rating}_{other\_node} + x$

Nodes store local ratings of encountered nodes, and exchange these ratings during

encounters. An encountered node's *trust score* is the sum of the local rating and foreign ratings. Upon receiving a message, the node checks if the forwarding node is the source of the message. If so, and if the *trust score* is not greater than the *trust threshold*, then the receiving node will discard the message and notify the forwarding node that it has been detected as selfish.

Nodes do not accept messages if they believe the source of the message to be selfish. To allow nodes that have been deemed as selfish to improve their trust score, nodes do pass messages to selfish nodes. If selfish nodes are given messages, they can then use the messages to improve their ratings. This approach does not punish nodes that are rarely given messages to forward, it only punishes those that could have given a message to a destination but did not. To prove that encounters took place we assume the presence of encounter tickets [89]. Nodes use this cryptographic mechanism to prove that they exchanged messages, by getting a signed receipt of message exchange.

While nodes do not need synchronised clocks, which can be difficult to implement in an opportunistic network, they must agree on the relative ordering of encounters. When nodes encounter one another they exchange the time they believe the current encounter is taking place at; nodes can thus determine the time when the encounters in the foreign history took place relative to their own opinion of the correct time. Nodes can then use this information despite potential differences in the perceived time on the nodes. A similar mechanism to IRONMAN could be used at the clock synchronisation layer to detect lies in the time dimension. We save this, however, for future work.

6.2 Evaluation

We test IRONMAN's performance using trace-driven simulation of a simple message-passing application. As the performance of an opportunistic network may vary depending on the connectivity patterns of the nodes, we use three real-world traces in our analysis. We believe that these traces capture a wide variety of possible network scenarios. To avoid the effects of warmup or ending segments of the trace where the recent/impending start/end of the trace may affect node behaviour, we divide the traces into segments and

use the most central segment:

1. Our "SASSY" connectivity trace (as discussed in Chapter 4.3). As the SASSY trace lasts over two months we split it into two 30-day segments and a 20 day segment and use the second segment.
2. The MIT Reality Mining (RM) trace [40] (as discussed in Chapter 4.3). The nine-month RM trace is divided into three 30-day periods, from the beginning, middle and end of the trace respectively, and we use the middle segment.
3. The HOPE dataset [1] (as discussed in Chapter 4.3). For the HOPE trace we use the second day of three.

We do not extract only day or nights, to allow for the heterogeneity of user behaviour. Even during the conference traces, only focussing on daylight hours would not allow for the nodes to exchange messages between hotel rooms for example. We do not wish to limit opportunistic communication by reducing the amount of time available for exchanging messages.

We also consider then length of trace that would be useful for evaluation. The time periods we use are appropriate as either the length of the traces are short begin with, as with the HOPE trace, or are datasets involving students. With datasets involving students we want to capture typical behaviour. We therefore look at the encounter frequency to try to avoid holiday periods where students would not be present in the same geographical area, so that we have sufficient encounters between the participants. We split the traces accordingly.

Table 4.1 shows the different properties of the overall traces and the SRSNs. For the RM and HOPE traces the number of edges, clustering coefficient and graph density are higher in the encounter data than in the SRSN data. In other words, there are longer paths in the SRSN data, which might make them useful for building a trust mechanism.

Nodes which are further away in the SRSN network might be less trusted, despite their frequent proximity in the trace network, reflecting the idea of familiar strangers [151]: nodes who you encounter frequently yet you do not know well.

6.2.1 Routing protocols

We analyse IRONMAN running over the Epidemic [142] routing protocol. We simulate performance at three different levels of selfishness following [122]: 0%, 50%, and 100%. Given Xu et al.'s finding [146] that the altruism of high-degree nodes is the most important for mitigating the impact of selfishness, we choose the nodes with the highest degree in the encounter graph to be selfish, which finds nodes that see the most number of different other nodes. This selection of selfish nodes maximises the impact of selfishness on our simulations.

Table 6.1: Simulation Routing Parameters

Parameter	Value (SASSY/RM/HOPE)
TTL of messages	10 days / 2 days / 2 hours
Message frequency	1 per node per day
Simulation length	30 days / 30 days / 1 day
Message size (MB)	1
Buffer size (MB)	2000
Loss per second (mAh)	1.9×10^{-6}
Time to send bundle (s)	34
Max energy (mAh)	1200
Energy per send (mAh)	0.4
Charge time (h)	8

6.2.2 Incentive mechanisms

We compare IRONMAN with two existing incentive mechanisms, and modifications of these two mechanisms to use SRSN information:

- IRONMAN: The mechanism outlined in Section 6.1. We use a value of 100 as the default local rating for SRSN nodes, 50 for unknown nodes and 50 for the trust threshold. We use 50 as the behavioural constant. With these values, nodes will

trust those members of their SRSN, but not other nodes. If another node provides them with a positive opinion of a third untrusted node with the default score of 50, the reputation score increase will be enough to raise opinion of the third node so that it is trusted.

- YSS: The peer-to-peer reputation mechanism developed by Yu, Singh and Sycara [152]. This scheme is appropriate for comparison with IRONMAN as it is a decentralised reputation mechanism that does not use any oracle or credit information. For the QoS parameter required to measure nodes' behaviour, we use the proportion of messages exchanged altruistically, detected using the same approach described in Section 6.1.1. Where possible we use the thresholds outlined in their paper: the default opinion of nodes that are not known is 0.5, which is the same value as the trust threshold. We use their exponential approach to weighting opinions and credibility of opinions.

- RELICS+S: A modified version of RELICS [139], as representative of the state-of-the-art in incentive mechanisms for opportunistic networks. This mechanism does not use any oracle information or any credit information. By using this approach RELICS is suitable for use in ad-hoc, decentralised opportunistic networks such as those used in the simulations in this work. We attempt to use parameters as described in their paper: 0.8 for the desired delivery ratio threshold, 0.373mAh as the initial energy level, one hour as the energy epoch, and 14.30mAh for the increase in energy allowed during each epoch. Nodes are given a starting rank of two, allowing them to send two messages before being required to forward on behalf of other nodes. Estimated delivery probabilities are 1.0 if a node is the source of a message, and 0.5 otherwise. As RELICS, unlike the other mechanisms, uses delivery receipts, we simulate these, but assume that receipts have no forwarding cost, to maximise the potential performance of RELICS. As RELICS does not actively detect selfishness, we modified the mechanism to treat nodes who have a rating below the initial value of two as selfish, we call this RELICS+S.

- YSS+SRSN: Here we bootstrap the mechanism so that nodes give members of their social network an opinion of 1.0 (complete trust). We leave the default trust level for other nodes as 0.5 and change the trust mechanism so that nodes only take the opinions of members of their social network into account.

- As a control, we also consider performance when no incentive mechanism is in place.

When a node is detected as selfish it ceases to behave selfishly. The node may continue to be punished, however, as other nodes need to detect its altruistic behaviour before trusting it again.

6.2.3 Scenarios and metrics

We perform simulations under two different scenarios to highlight different features of the mechanisms:

1. A scenario with no resource constraints: nodes have infinite buffer space and energy and messages have an infinite time-to-live (TTL) value. There will be no false positives of selfishness generated in this scenario, as nodes will not drop messages due to full buffers.

2. Finite buffers, energy and TTL, as listed in Table 6.1.

We simulate ten runs per scenario, per incentive mechanism, per trace. We calculate the delivery ratio (number of delivered delivered over messages sent) and observe the difference in performance across incentive mechanisms.

15% of each trace is used as a warmup period, where no messages are created or sent, but nodes may build up reputation information. All message senders and destinations are picked from an exponential distribution. For the SASSY and RM traces, we exponentially distribute the message creation times throughout the day. As the HOPE trace only lasts one day, we uniformly distribute the message creation times throughout the day to prevent messages from being created with no time left in the trace for them to be sent.

As nodes in a real deployment would have memory limits, we restrict the size of the history of recent encounters for IRONMAN and both YSS mechanisms, to the most recent 1000 entries in all scenarios.

To study the performance of the incentive mechanisms we consider the negative impact of selfishness on the network using the following metrics:

1. *Detection Time*. The time that it takes a mechanism to correctly detect selfish behaviour in a node. This is the average time between a node choosing to behave selfishly, and the time that a node is detected as selfish. A mechanism with a low detection time will minimise the impact selfish behaviour has on the network.

2. *Detection Accuracy*. The proportion of selfish nodes that were correctly detected as selfish by a mechanism. An ideal mechanism will have a low Detection Time and a high Detection Accuracy.

3. *Selfishness Cost*. The proportion of forwarded messages (medium accesses) that were generated as a result of a node creating a message while it was selfish. In some respects this can be seen as the "goodput" of a network with selfish nodes; a mechanism with a low Selfishness Cost is effectively maximising the use of the network by cooperative nodes.

6.3 Results

We now examine the performance of the incentive mechanisms in our simulations.

6.3.1 Infinite buffer, energy and TTL scenario

First we consider the performance of the routing protocol when nodes have infinite battery and energy resources, and the message have an infinite TTL. Figures 6.3a–6.3c show network performance (in terms of delivery ratio) across the three traces. It can be seen that IRONMAN performs the best of the evaluated mechanisms. All the mechanisms have quite high detection times, due to intermediate nodes infrequently encountering destination nodes, but IRONMAN has a higher detection accuracy and lower detection time than all the other mechanisms in the SASSY and RM traces (Figures 6.4a and 6.4b). In the much denser HOPE trace, even though the detection accuracy is lower than the YSS-based

mechanisms (Figure 6.4c), the resulting delivery ratio (Figure 6.3c) is higher because both YSS-based mechanisms discard more messages from selfish nodes. Indeed, in this trace (Figure 6.3c), IRONMAN is not only the sole mechanism that performs better than having no detection mechanism at all, but IRONMAN performs almost as well with 100% of nodes acting selfish as when 0% of nodes are selfish.

In addition to the fastest and most accurate detection of selfishness, IRONMAN has a lower or equivalent selfishness cost than the other mechanisms (Figures 6.5a–6.5c). In other words, IRONMAN is successful at ensuring that the network is predominantly used by cooperating nodes.

When the YSS mechanism is modified to use social network information, it performs the same, or better, than the original mechanism (Figures 6.3a–6.3c). Figures 6.4a–6.4c show that while YSS+SRSN has a slower detection time than YSS in two of the traces, it has a higher delivery ratio, as it does not drop as many messages from nodes perceived as selfish.

6.3.2 Resource-constrained scenario

As one might expect, when we consider the effects of energy, buffer and TTL, we see that in comparison to the infinite resource scenario, the network performance drops. In the RM and HOPE traces (Figures 6.6b–6.6c), IRONMAN has the highest delivery ratios, while in the SASSY trace (Figure 6.6a), all mechanisms perform similarly. IRONMAN, however, detects a larger proportion of selfish nodes (Figure 6.7a), has a lower detection time and a lower selfishness cost (Figure 6.8a).

The relative detection time is not consistent across the traces, however. YSS and YSS+SRSN have a lower detection time and higher detection accuracy in the HOPE trace (Figure 6.7c) than IRONMAN; the density of the HOPE trace means the low detection time of YSS and YSS+SRSN (a result of the exponential weightings of ratings/opinions) causes both mechanisms to detect a higher proportion of nodes than both IRONMAN and RELICS+S, and results in a lower detection time. As in the infinite scenario, however, IRONMAN still has a higher delivery ratio (Figure 6.6c), despite lower accuracy and detection time.

In the HOPE trace, the delivery ratios of YSS and YSS+SRSN do not change as selfishness in the network increases. These mechanisms are able to detect selfishness and drop messages from those selfish nodes, thereby reducing the overall performance of the network. YSS and YSS+SRSN have similar Selfishness Cost results to IRONMAN however, as YSS and YSS+SRSN do not allow nodes that have become altruistic to forward many messages. If a group of nodes all believe that each other are selfish, they will drop the messages created by the other nodes in the group. The only way to build up a good reputation is to forward messages for other nodes, if nodes drop all incoming messages they can not build up enough reputation to have their own messages forwarded. The delivery ratio therefore remains low, and the Selfishness Cost remains high, as is the case for YSS and YSS+SRSN in the HOPE trace.

Again we see that RELICS+S does not perform as well as IRONMAN, with a lower delivery ratio (Figures 6.6a–6.6c). This is because the energy monitor does not allow for nodes to forward a sufficient amount of messages. RELICS+S has a low detection accuracy in all traces (Figures 6.7a–6.7c), and a high detection time in all but the SASSY trace. This is because RELICS+S does not detect selfishness well enough, a problem exacerbated by reduced forwarding opportunities.

Figures 6.8a–6.8c show that IRONMAN has the lowest selfishness cost in all traces; IRONMAN is again the best at reducing the overall impact of selfish nodes on the network.

Overall we see that IRONMAN can perform well, in certain scenarios performing as when there are no selfishness nodes in the network. SRSN-based mechanisms are always the best (or equivalent to) the best performing mechanism in the network. The exception is RELICS+S, which continues to perform badly, because the time to adjust to the correct energy level causes nodes to miss out on forwarding opportunities.

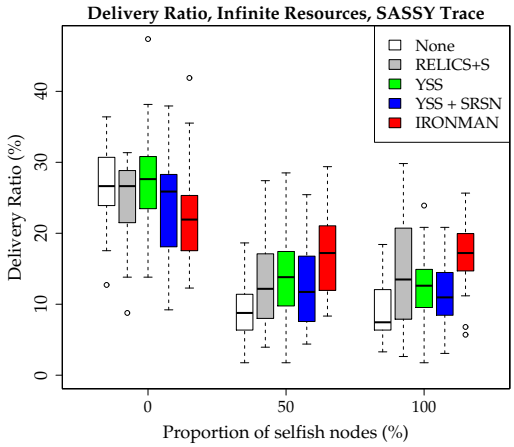
6.4 Conclusions And Future Work

We have introduced IRONMAN, an incentive mechanism for opportunistic networks that uses pre-existing social-network information to bootstrap trust relationships. Unlike ex-

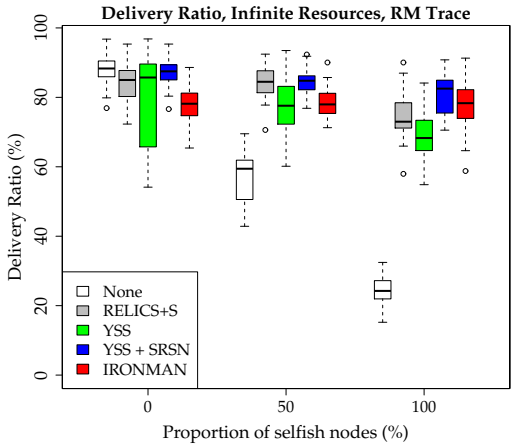
isting mechanisms, IRONMAN does not require an oracle or infrastructure network, nor delivery receipts.

We have demonstrated that IRONMAN outperforms existing incentive mechanisms, with accurate detection of selfish nodes in a timely manner, and improved delivery performance in the presence of selfishness. As a result, IRONMAN is able to maximise the proportion of the network that is used by cooperating nodes.

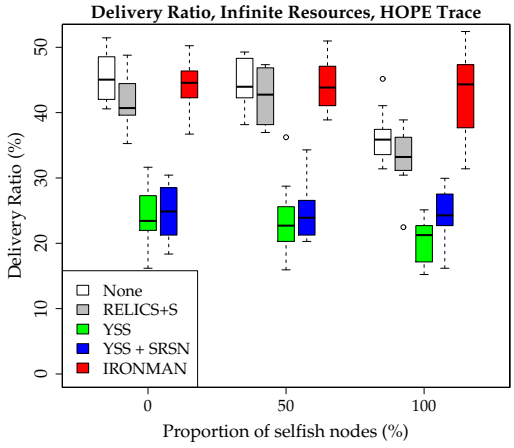
We have also shown that social-network information can be used to improve existing incentive mechanisms in a similar manner. We believe that this use of social-network information will prove a fruitful topic for researchers in this and other networking areas. For instance, is it possible to use social-network information to improve incentive mechanisms for peer-to-peer or ad hoc networks?



(a) In the SASSY trace, all the mechanisms perform similarly. IRONMAN and RELICS+S have the best performance when 100% of the nodes are selfish; however, IRONMAN has less variance.

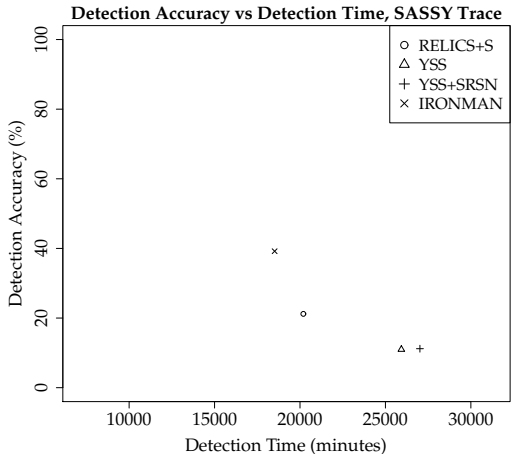


(b) In the RM trace, all the mechanisms perform similarly, with YSS performing slightly better than the others at high levels of selfishness.

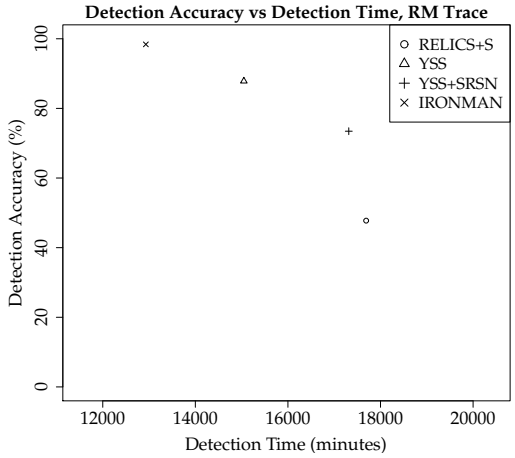


(c) In the HOPE trace, IRONMAN performs equivalently to having no selfish nodes in the network. Other mechanisms detect selfishness but do not allow enough forwarding: YSS and YSS+SRSN drop messages from nodes they detect as selfish, and RELICS+S's energy monitor allows too little forwarding.

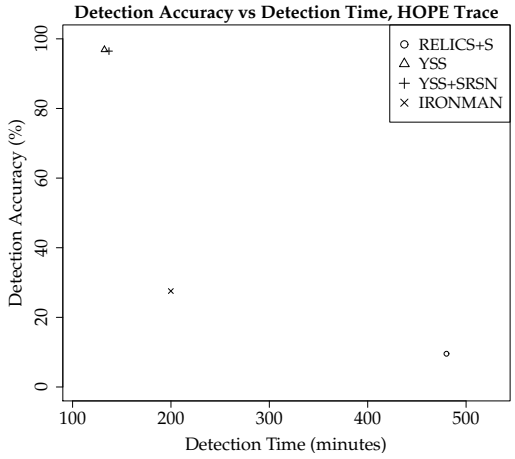
Figure 6.3: Incentive mechanism performance in the three traces under epidemic routing, with infinite buffer, energy and TTL.



(a) In the SASSY trace, IRONMAN performs the best, with the highest accuracy in the lowest time.

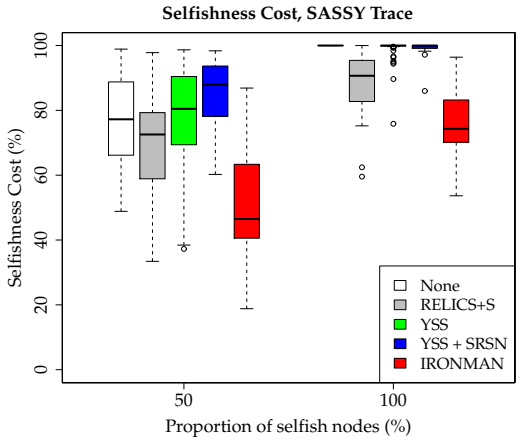


(b) In the RM trace, IRONMAN again performs best. YSS performs better than YSS+SRSN as it does not implicitly trust as many nodes.

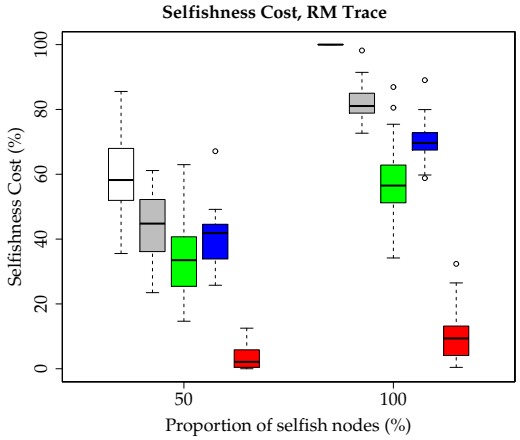


(c) In the HOPE trace, YSS+SRSN and YSS perform best, with low detection time and high accuracy. In spite of this, IRONMAN still has the highest delivery ratio (Figure 6.6c).

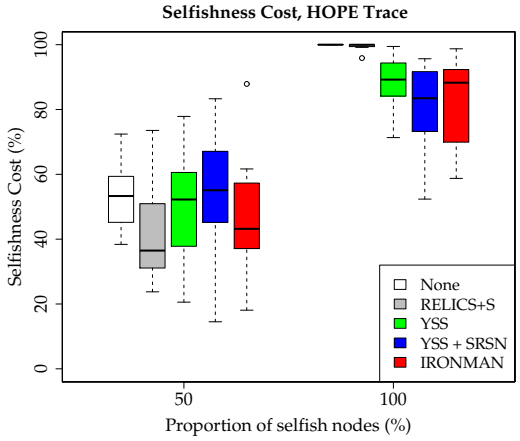
Figure 6.4: Detection Accuracy against detection time, when 100% of nodes are selfish. Infinite buffer, energy and TTL.



(a) In the SASSY trace, IRONMAN performs best, as its low detection time ensures that more nodes are incentivised away from selfishness before sending messages.

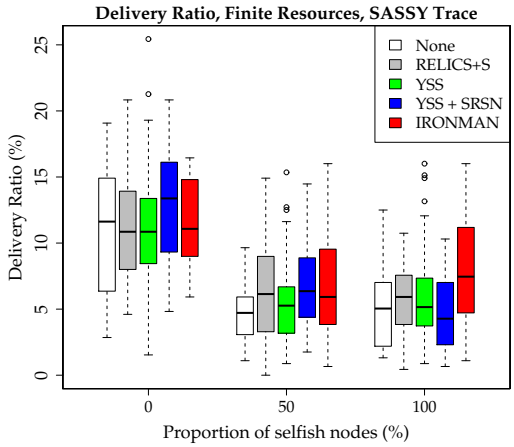


(b) In the RM trace, IRONMAN performs very well, ensuring that almost all medium accesses are from altruistic nodes. Legend as in Figure 6.5a.

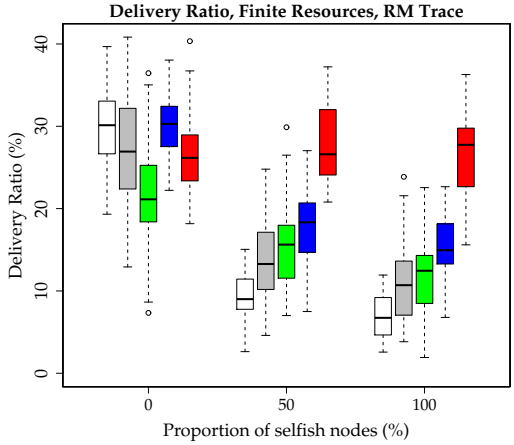


(c) In the HOPE trace, all mechanisms perform similarly apart from when all nodes in the network are selfish. The energy model in RELICS penalises altruistic nodes.

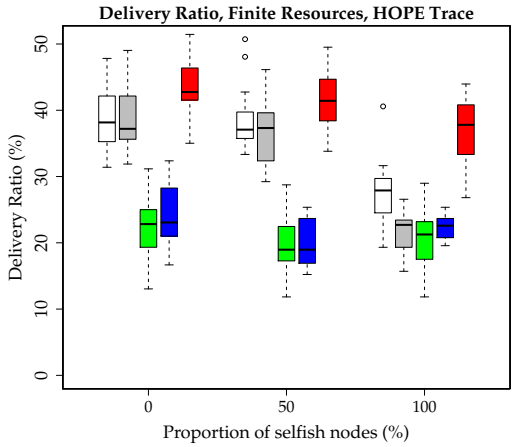
Figure 6.5: Selfishness Cost under epidemic routing and infinite buffer, energy and message TTLs.



(a) In the SASSY trace, all mechanisms perform similarly, with IRONMAN performing slightly better at 100% selfishness.

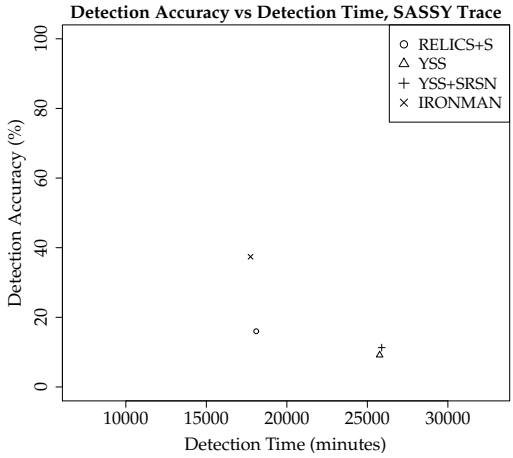


(b) In the RM trace, IRONMAN performs far better than the other mechanisms, performing as well as having no selfish nodes in the network. Legend as in Figure 6.6a.

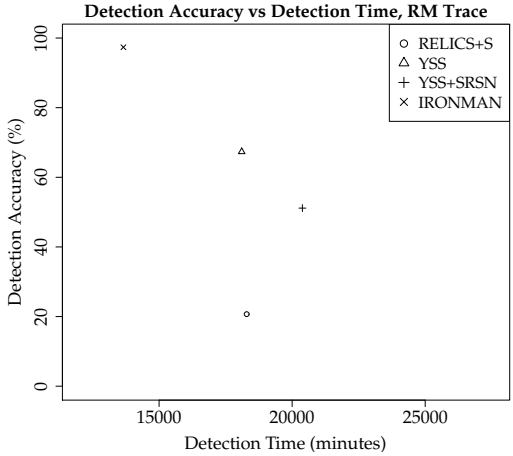


(c) The level of selfishness does not affect normal delivery, however, almost all mechanisms apart from IRONMAN do not perform well at 100% selfishness. Legend as in Figure 6.6a.

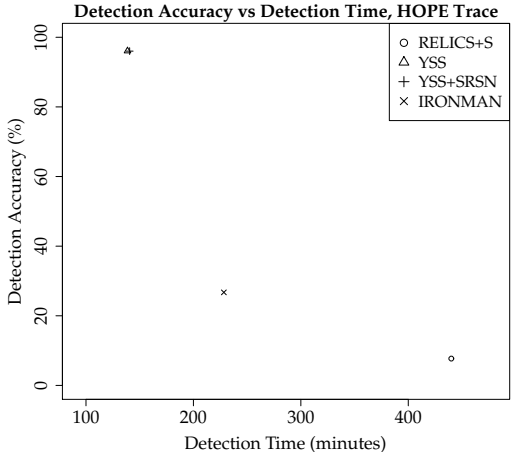
Figure 6.6: Incentive mechanism performance under epidemic routing, with finite buffer, energy and TTL.



(a) In the SASSY trace, IRONMAN provides the highest accuracy in the lowest time.

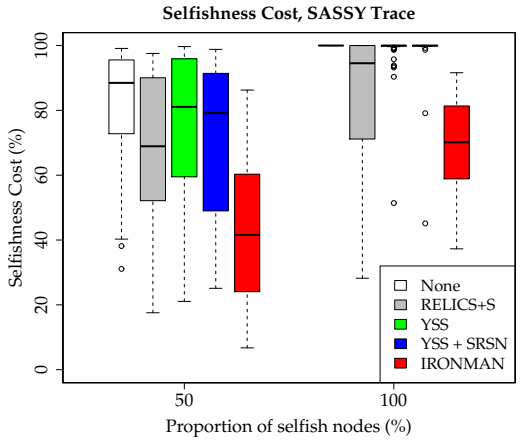


(b) In the RM trace, IRONMAN again has the highest accuracy in the quickest time.

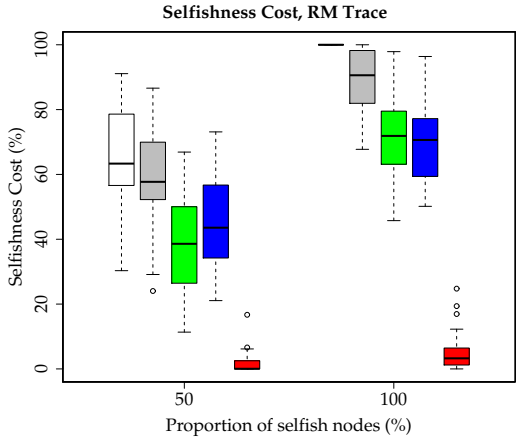


(c) In the HOPE trace, YSS and YSS+SRSN perform best in the dense HOPE trace.

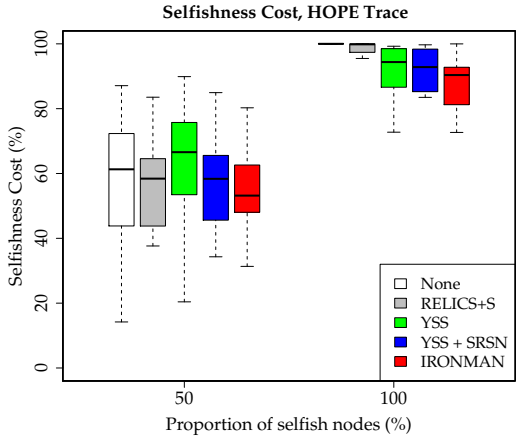
Figure 6.7: Detection Accuracy against detection time, when 100% of nodes are selfish. Finite TTL, buffer and energy.



(a) In the SASSY trace, IRONMAN performs well, with the lowest selfishness cost.



(b) In the RM trace, IRONMAN greatly outperforms the other mechanisms, with both YSS mechanisms performing similarly, followed by RELICS. Legend as in Figure 6.8a.



(c) In the HOPE trace, all mechanisms perform similarly, apart from when all nodes are selfish, when RELICS performs almost as badly as having no detection mechanism.

Figure 6.8: Selfishness Cost under epidemic routing and finite buffer, energy and TTL.

Chapter 7

Conclusions

Opportunistic routing relies on the interactions of participants to exchange data. Such interactions arise because of social behaviour; the study of social networks can therefore be useful for opportunistic routing. Many existing opportunistic routing protocols rely on encounter histories to make forwarding decisions. We therefore addressed the following thesis:

> Self-reported social networks provide an alternative to encounter histories for efficient routing in opportunistic networks.

To test the thesis we answered the following questions:

1. How can we bootstrap an opportunistic network?
2. How can we incentivise participation in opportunistic DTNs?

To address the first question we have shown (in Chapters 4 and 5) that self-reported social networks can provide an alternative to encounter histories. We demonstrated this by running collecting encounter data and combining these with existing traces and performing trace-driven simulation of opportunistic routing.

First, in Chapter 4, we demonstrated that SRSNs can be used for routing and have a lower cost than using encounter data for routing. Second, in Chapter 5, we demonstrated

that by analysing the roles that nodes play in their SRSN we can build a routing protocol that performs better than encounter history based protocols for bootstrapping an opportunistic network.

To address the second question, in Chapter 6, we demonstrated that SRSNs can be used to build an incentive mechanism for opportunistic routing that detects selfish messaging behaviour and punishes offending nodes, thereby encouraging participation.

We now outline the contributions of this thesis. We discuss how this relates to work in the area, and future research directions effected by this thesis.

7.1 Contributions

In Chapter 4 we analysed the performance of an opportunistic routing protocol where nodes may only forward a message if the encountered node is in a subset of nodes defined by the message source. The subset was either the SRSN of the node, or the DSN of the node. The DSN was taken to be any node seen by the node during the trace, making use of future information.

We used three traces to drive our simulation, and observe that in certain scenarios, the SRSN has a lower delivery cost than the DSN, while providing an equivalent delivery ratio. In scenarios where the SRSN does not have an equivalent delivery ratio, the SRSN also displays different properties of delivery cost. We believe that the application designer must decide which protocol is appropriate for routing in a given scenario. We have shown then, that SRSNs can be useful for routing, indicating that it is possible to effectively route information better than, or equivalent to, DSNs.

In Chapter 5, we developed and analysed an opportunistic routing protocol, SRR, that uses social roles to drive nodes' routing decisions. Many existing routing protocols use encounter histories: nodes use the data gathered from the encounters they participate in. SRR uses the social roles derived from a node's SRSN to decide whether a node can be given a message. Using trace driven simulation on four encounter traces, we demonstrated that SRR performs as well as existing protocols for message routing, however, it always

performs better to bootstrap the network.

We also see that in dense traces, despite its high cost, SRR has a lower standard deviation in the distribution of cost amongst the network nodes. This shows that SRR avoids adversely overloading nodes that perform a large bulk of the forwarding.

Thus, we have demonstrated the usefulness of SRSN information for opportunistic routing in comparison to existing routing protocols, and specifically to encounter history based protocols.

In Chapter 6, we studied how to incentivise the nodes in an opportunistic network to follow the routing protocol. We built an incentive mechanism for opportunistic networks IRONMAN. IRONMAN does not rely on infrastructure networks, delivery receipts or oracles with perfect information. Instead IRONMAN uses SRSN information to bootstrap the incentive protocol, and uses node encounters to exchange reputation information. We also demonstrated that we can use SRSN information to improve existing incentive mechanisms, by using the SRSN information in conjunction with an existing protocol. We compared the protocols using trace-driven simulation, using three traces.

7.2 Discussion

This thesis shows the usefulness of SRSN information in two areas of opportunistic networking: routing protocols and incentivising participation in routing.

In Chapter 4, we treat an encounter as an indicator of a tie in the DSN. We could, however, compute the ties using a different approach. Perhaps one tie is not enough contact to indicate a social-network association?

We also used traces of various sizes, both in terms of the number of nodes in the network, and the number of encounters in the networks. We chose this variety of sizes to allow for variance in behaviour across differing network scenarios. We considered using longer traces that take place over many months, we believe however, that we would not see any relevant difference in results.

The length of time used for the warmup period in Chapter 5 is obtained from existing work. We could perhaps vary the length of the warmup period to measure the effect on the relative performance of the routing protocols.

As the NUS trace is so large, we cut down the number of nodes using methodology used in related work. This approach maintains the same degree-distribution of the nodes, which we believe is appropriate for use in our work. We could, however, attempt to see what effect using all the nodes in the trace makes. As well as the number of nodes, we assumed that a portion of the nodes would be absent from the trace in each day of the trace. We assumed that this value was normally distributed.

In Chapter 5 we introduced SRR to demonstrate the use of social roles derived from SRSNs. This is a relatively simplistic protocol, we could imagine an approach able to transition from using the SRSN to using the DSN for role routing. We could also improve SRR by limiting the number of message duplications in some way.

We demonstrated that SRR shows a lower standard-deviation in the distribution of forwarding in dense encounter traces. We could study this property further and look at metrics for detecting unfairness, or the effects of routing protocols at the node level, rather than across the network.

We presented IRONMAN, an incentive mechanism for opportunistic networks. It uses node encounters to exchange reputation information that leads to the distribution of knowledge of node selfishness. Nodes using IRONMAN do not prioritise the messages in the outbound buffer. Perhaps prioritising messages from nodes in the SRSN would further incentivise participation.

We hope that the use of SRSN information will become more common in opportunistic networks and other networking areas. Using SRR to demonstrate that opportunistic networks are able to effectively route messages from network startup encourages the adoption of opportunistic networks.

We have shown that in comparison to many of the existing incentive mechanisms, IRONMAN achieves better performance without the need for oracles, infrastructure net-

works or delivery receipts by using SRSN information.

7.3 Future Work

We have found that there is space to research into metrics that capture opportunistic routing protocol effects on individual nodes (as opposed to the network level). We believe our results show that an understanding of these effects is important for building effective opportunistic routing protocols.

Routing protocols that make more use of social roles could be useful for providing improved opportunistic routing. This is especially true to attempt to maintain more even forwarding distributions which may be considered *fairer*. Rather than viewing the problem of nodes being frequently selected for forwarding as an issue related to delivery cost, a concept of fairness for opportunistic routing could provide more developments for incentive mechanisms for opportunistic networks.

We could perform user studies into the privacy concerns and altruism of opportunistic network participants. For example, users may not wish to forward messages for individuals that they do not consider themselves to have a strong social connection to. Privacy concerns of users may include:

- not wanting their messages forwarded through "non-friends", regardless of whether messages were encrypted
- concerns over their reputation/opinion values from the incentive mechanism being distributed amongst nodes in the network, friends or otherwise
- not trusting other individuals reputation/opinion values, and instead only choosing to consider the opinions of friends

If we can identify and address user concerns, we hope that we can motivate the use of opportunistic networks more effectively.

We have demonstrated that the choice of routing protocol during the warmup period

is important. We could ask which is the best protocol to use after the warmup period. We assume that the choice of protocol depends on the nature of the opportunistic network. We could look into how nodes detect which routing protocol to use, and on what grounds that decision could be made. Can an opportunistic network work in the face of multiple routing protocols being in use simultaneously?

Our plans for future work include analysing a network's connectivity and structural properties during the warm-up period to determine which routing protocol is the most appropriate to use after the warm-up period, investigating the affect of varying the minimum role size, and further comparison of role-graphs and self-reported social networks.

We could explore the interaction between the application-layer social-network information that we exploit for our incentive mechanism, and the use of this information in the application itself. Many opportunistic network applications might themselves involve social networks, for instance, mobile social networks, crowdsourcing, or participatory sensing. It may be useful to expose trust relationships from the routing layer to the application layer, or vice versa. Could application-layer detection of misbehaving nodes, such as anomalous crowdsourced data, be used to inform routing decisions? Such further study requires both routing protocol development and application deployment.

We also wish to refine our models of social network behaviour. We currently assume that members of the same social network will be more likely to trust each other. If behaviour is contagious across a social network, as proposed by Fowler and Christakis [51], then perhaps selfish behaviour might also propagate, leading to new incentive challenges.

Appendix A

Social network Vocabulary

We take the vocabulary from [52], and expand with relevant entries:

Actor: A node or vertex

Dyad/tie: A relation between two actors

Neighbours: Two nodes are neighbours if they connect via a tie

Degree: The number of ties that a node has

Clique: A subset of actors with ties with all other members of the subset

Density: The proportion of the total available ties connecting actors

Centralization: The fraction of main actors within a network

Reachability: The number of ties connecting actors

Connectedness: The ability of actors to reach one another reciprocally

Asymmetry: The ratio of reciprocal relationships— those relationships that are mutual– to total relationships within a network

Balance: The extent to which ties in the network are direct and reciprocated

Centrality: The degree to which an actor is in a central position in the network

Isolate: An actor with no ties to other actors

Gatekeeper: An actor who connects the network to outside influences

Cutpoint: An actor whose removal results in unconnected paths in the network

Ego-network: A graph consisting of the connections from one node to its neighbours, and the connections between those neighbours

Diameter: The longest shortest-path between any two nodes in the network

Bibliography

[1] Aestetix and Christopher Petro. CRAWDAD data set hope/amd (v. 2008-08-07). Downloaded from http://crawdad.cs.dartmouth.edu/hope/amd, August 2008.

[2] Yong Y. Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 835–844, New York, NY, USA, 2007. ACM.

[3] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Internet: Diameter of the World-Wide web. *Nature*, 401(6749):130–131, September 1999.

[4] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[5] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences*, 97(21):11149–11152, October 2000.

[6] Xueli An, Jing Wang, Venkatesha R. Prasad, and I. G. M. M. Niemegeers. OPT: online person tracking system for context-awareness in wireless personal network. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 47–54, New York, NY, USA, 2006. ACM.

[7] T. Anantvalee and J. Wu. Reputation-Based system for encouraging the cooperation of nodes in mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications, 2007. ICC '07.*, pages 3383–3388, June 2007.

[8] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, 2007.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[10] Russell H. Bernard, Peter D. Killworth, and Lee Sailer. Informant accuracy in social-network data v. an experimental attempt to predict actual communication from recall data. *Social Science Research*, 11(1):30–66, March 1982.

[11] Greg Bigwood and Tristan Henderson. Bootstrapping opportunistic networks using social roles. *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–6, June 2011.

[12] Greg Bigwood and Tristan Henderson. IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. In *Proceedings of the 3rd IEEE Conference on Social Computing (SocialCom 2011) (to appear)*, MIT, Boston, USA, October 2011.

[13] Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti. Exploiting self-reported social networks for routing in ubiquitous computing environments. In *Proc. IEEE SAUCE*, Avignon, France, October 2008.

[14] Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti. CRAWDAD data set st_andrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.cs.dartmouth.edu/st_andrews/sassy, June 2011.

[15] C. Boldrini, M. Conti, F. Delmastro, and A. Passarella. Context- and social-aware middleware for opportunistic networks. *Journal of Network and Computer Applications*, 33(5):525–541, September 2010.

[16] Chiara Boldrini, Marco Conti, and Andrea Passarella. Social-based autonomic routing in opportunistic networks. In Athanasios V. Vasilakos, Manish Parashar, Stamatis Karnouskos, and Witold Pedrycz, editors, *Autonomic Communication*, chapter 2, pages 31–67. Springer US, Boston, MA, 2009.

[17] S. Borgatti. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, December 1993.

[18] Sonja Buchegger and Jean Y. Le Boudec. Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 226–236, New York, NY, USA, 2002. ACM.

[19] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant networks. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–11. IEEE, April 2006.

[20] John Burgess, George D. Bissias, Mark D. Corner, and Brian 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, pages 61–70, New York, NY, USA, 2007. ACM.

[21] 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. *Communications Magazine, IEEE*, 41(6):128–136, 2003.

[22] Ronald S. Burt. Positions in networks. *Social Forces*, 55(1):93–122, 1976.

[23] Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, and Ronald A. Peterson. People-centric urban sensing. In *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*, page 18, New York, NY, USA, 2006. ACM.

[24] J. Candia, M. C. González, P. Wang, T. Schoenharl, G. Madey, and A. Barabási. Uncovering individual and collective human dynamics from mobile phone records. *ArXiv e-prints*, 710, October 2007.

[25] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), April 2007.

[26] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[27] Augustin Chaintreau, Abderrahmen Mtibaa, Laurent Massoulie, and Christophe Diot. The diameter of opportunistic mobile networks. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM.

[28] J. Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. *Proceedings of 37-th Annual Allerton Conference on Communication,Control, and Computing*, September 1999.

[29] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). http://www.ietf.org/rfc/rfc3626.txt, October 2003.

[30] Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. From opportunistic networks to opportunistic computing. *IEEE Communications Magazine*, 48(9):126–139, September 2010.

[31] Mauro Conti, Roberto Di Pietro, Andrea Gabrielli, Luigi V. Mancini, and Alessandro Mei. The smallville effect: social ties make mobile networks more secure against node capture attack. In *Proceedings of the 8th ACM international workshop on Mobility management and wireless access*, MobiWac '10, pages 99–106, New York, NY, USA, 2010. ACM.

[32] Steven R. Corman and Lisa Bradford. Situational effects on the accuracy of Self-Reported organizational communication behavior. *Communication Research*, 20(6):822–840, December 1993.

[33] Dana Cuff, Mark Hansen, and Jerry Kang. Urban sensing: out of the woods. *Commun. ACM*, 51(3):24–33, 2008.

[34] David E. Culler and Wei Hong. Wireless sensor networks: Introduction. *Commun. ACM*, 47:30–33, June 2004.

[35] 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 2008.

[36] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *MobiHoc '07: Proceedings of the 8th ACM interna-*

tional symposium on Mobile ad hoc networking and computing, pages 32–40, New York, NY, USA, 2007. ACM Press.

[37] R. Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, June 1992.

[38] R. Dunbar. Coevolution of neocortex size, group size and language in humans. *Behavioral and Brain Sciences*, 16(4):681–735, 1993.

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

[40] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

[41] Nathan Eagle, Alex S. Pentland, and David Lazer. Inferring social network structure using mobile phone data, 2008.

[42] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale-free topology of e-mail networks. *Physical Review E 66 (2002) 035103(R)*, February 2002.

[43] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G. S. Ahn, and A. T. Campbell. The BikeNet mobile sensing system for cyclist experience mapping. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 87–101, New York, NY, USA, 2007. ACM.

[44] Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Gahng S. Ahn, and Andrew T. Campbell. MetroSense project: People-Centric sensing at scale. *World-Sensor-Web at SenSys*, October 2006.

[45] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[46] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. Delegation forwarding. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 251–260, New York, NY, USA, 2008. ACM.

[47] Flavio Fabbri and Roberto Verdone. A sociability-based routing scheme for delay-tolerant networks. *EURASIP J. Wirel. Commun. Netw.*, 2011, January 2011.

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

[49] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, volume 29 of *SIGCOMM '99*, pages 251–262, New York, NY, USA, October 1999. ACM.

[50] Office for National Statistics. National statistics online - consumer durables. urlhttp://www.statistics.gov.uk/CCI/nugget.asp?ID=868&Pos=3&ColRank=2&Rank=240, November 2010.

[51] James H. Fowler and Nicholas A. Christakis. Cooperative behavior cascades in human social networks. *Proceedings of the National Academy of Sciences*, 107(12):5334–5338, March 2010.

[52] Kimberly A. Fredericks and Maryann M. Durland. The historical evolution and basic concepts of social network analysis. *New Directions for Evaluation*, 2005(107):15–23, 2005.

[53] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, March 1977.

[54] Noah E. Friedkin. Structural cohesion and equivalence explanations of social homogeneity. *Sociological Methods & Research*, 12(3):235–261, February 1984.

[55] Saurabh Ganeriwal, Laura K. Balzano, and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, 4(3):1–37, June 2008.

[56] Deepak Ganesan, Ben Greenstein, Denis Perelyubskiy, Deborah Estrin, and John Heidemann. An evaluation of multi-resolution storage for sensor networks. In

SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, pages 89–102, New York, NY, USA, 2003. ACM Press.

[57] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Multicasting in delay tolerant networks: a social network perspective. In *MobiHoc '09*, MobiHoc '09, pages 299–308, New York, NY, USA, 2009. ACM.

[58] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[59] Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–null, May 1973.

[60] M. Grossglauser and M. Vetterli. Locating nodes with EASE: Last encounter routing for ad hoc networks through mobility diffusion, 2003.

[61] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, 2002.

[62] Matthias Grossglauser and Martin Vetterli. Locating mobile nodes with EASE: learning efficient routes from encounter histories alone. *IEEE/ACM Trans. Netw.*, 14(3):457–469, 2006.

[63] Hani Hagras. Embedding computational intelligence in pervasive spaces. *IEEE Pervasive Computing*, 6(03):85–89, 2007.

[64] Qi He, Dapeng Wu, and P. Khosla. SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks. In *Proceeding of the Wireless Communications and Networking Conference, WCNC. 2004*, pages 825–830 Vol.2, 2004.

[65] Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre. Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing. In *INFOCOM, 2010 Proceedings IEEE*, 2010.

[66] Wei J. Hsu, Debojyoti Dutta, and Ahmed Helmy. Profile-cast: behavior-aware mobile networking. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):52–54, January 2008.

[67] Elgan Huang, Jon Crowcroft, and Ian Wassell. Rethinking incentives for mobile ad hoc networks. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice*

and theory of incentives in networked systems, pages 191–196, New York, NY, USA, 2004. ACM.

[68] Bernardo A. Huberman and Lada A. Adamic. Internet: Growth dynamics of the World-Wide web. *Nature*, 401(6749):131, September 1999.

[69] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251, 2005.

[70] Pan Hui and Jon Crowcroft. Bubble rap: Forwarding in small world DTNs in ever decreasing circles. Technical Report UCAM-CL-TR-684, University of Cambridge, Computer Laboratory, May 2007.

[71] Pan Hui and Jon Crowcroft. How small labels create big improvements. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 65–70, Washington, DC, USA, 2007. IEEE Computer Society.

[72] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 241–250, New York, NY, USA, 2008. ACM.

[73] Pan Hui, A. Lindgren, and J. Crowcroft. Empirical evaluation of hybrid opportunistic networks. *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pages 1–10, January 2009.

[74] Pan Hui, Eiko Yoneki, Shu Y. Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *Proc. ACM MobiArch Workshop 2007*, pages 1–8, New York, NY, USA, 2007. ACM.

[75] Stratis Ioannidis and Augustin Chaintreau. On the strength of weak ties in mobile social networks. In *Proceedings of Second ACM Workshop on Social Network Systems*, March 2009.

[76] D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[77] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li S. Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 96–107, New York, NY, USA, 2002. ACM.

[78] Paris C. Kanellakis and Scott A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. In *PODC '83: Proceedings of the second annual ACM symposium on Principles of distributed computing*, pages 228–240, New York, NY, USA, 1983. ACM.

[79] Merkourios Karaliopoulos. Assessing the vulnerability of DTN data relaying schemes to node selfishness. *IEEE Communications Letters*, 13(12):923–925, December 2009.

[80] Ari Keränen, Mikko Pitkänen, Mikko Vuori, and Jörg Ott. Effect of non-cooperative nodes in mobile DTNs. In *Twelfth IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2011)*, 2011.

[81] Abdelmajid Khelil, Pedro J. Marron, and Kurt Rothermel. Contact-Based mobility metrics for Delay-Tolerant ad hoc networking. In *MASCOTS '05: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 435–444, Washington, DC, USA, 2005. IEEE Computer Society.

[82] M. Kilduff and W. Tsai. Social networks and organizations. *Human Resource Development Quarterly*, 16(4):557–561, 2005.

[83] Young B. Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[84] David Kotz and Tristan Henderson. CRAWDAD: A community resource for archiving wireless data at dartmouth. *IEEE Pervasive Computing*, 4(4):12–14, October 2005.

[85] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tompkins, and Eli Upfal. The web as a graph. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '00, pages 1–10, New York, NY, USA, 2000. ACM.

[86] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, April 2010. ACM.

[87] Sung J. Lee, William Su, and Mario Gerla. Wireless ad hoc multicast routing with mobility prediction. *Mob. Netw. Appl.*, 6(4):351–360, 2001.

[88] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Proceedings of Pervasive 2006*, pages 1–16, Dublin, Ireland, May 2006.

[89] F. Li, J. Wu, and A. Srinivasan. Thwarting blackhole attacks in Disruption-Tolerant networks using encounter tickets. In *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pages 2428–2436. IEEE, April 2009.

[90] Na Li and Sajal K. Das. RADON: reputation-assisted data forwarding in opportunistic networks. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, MobiOpp '10, pages 8–14, New York, NY, USA, February 2010. ACM.

[91] Na Li and Sajal K. Das. A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks*, February 2011.

[92] Fredrik Liljeros, Christofer R. Edling, Luis A. Amaral, H. Eugene Stanley, and Yvonne Aberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, June 2001.

[93] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[94] Anders Lindgren and Pan Hui. The quest for a killer app for opportunistic and delay tolerant networks: (invited paper). In *CHANTS '09: Proceedings of the 4th ACM workshop on Challenged networks*, pages 59–66, New York, NY, USA, 2009. ACM.

[95] Cong Liu and Jie Wu. Routing in a cyclic mobispace. In *MobiHoc '08*, MobiHoc '08, pages 351–360, New York, NY, USA, May 2008. ACM.

[96] Jinshan Liu and Valérie Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In Christian Jensen, Stefan Poslad, and Theo Dimitrakos, editors, *Trust Management*, volume 2995 of *Lecture Notes in Computer Science*, chapter 5, pages 48–62. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2004.

[97] Zhengye Liu, Hao Hu, Yong Liu, Keith W. Ross, Yao Wang, and Markus Mobius. P2P trading in social networks: the value of staying connected. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 2489–2497, Piscataway, NJ, USA, 2010. IEEE Press.

[98] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xuemin Shen, and Bruno Preiss. Pi: A practical incentive protocol for delay tolerant networks. *IEEE Transactions on Wireless Communications*, 9(4):1483–1493, April 2010.

[99] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 231–244, Berkeley, CA, USA, 2005. USENIX Association.

[100] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 255–265, New York, NY, USA, 2000. ACM.

[101] Afra J. Mashhadi, Sonia B. Mokhtar, and Licia Capra. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In *WoWMoM '09*, June 2009.

[102] Liam McNamara, Cecilia Mascolo, and Licia Capra. Media sharing based on colocation prediction in urban transport. In *Proceedings of the 14th ACM international confer-*

ence on Mobile computing and networking, MobiCom '08, pages 58–69, New York, NY, USA, 2008. ACM.

[103] Pietro Michiardi and Refik Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

[104] Andrew G. Miklas, Kiran K. Gollu, Kelvin K. W. Chan, Stefan Saroiu, Krishna P. Gummadi, and Eyal De Lara. Exploiting social interactions in mobile systems. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 409–428, Berlin, Heidelberg, 2007. Springer-Verlag.

[105] Mehul Motani, Vikram Srinivasan, and Pavan S. Nuggehalli. PeopleNet: engineering a wireless virtual social network. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 243–257, New York, NY, USA, 2005. ACM.

[106] Abderrahmen Mtibaa, Augustin Chaintreau, Jason Lebrun, Earl Oliver, Anna K. Pietilainen, and Christophe Diot. Are you moved by your social network application? In *Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, Seattle, WA, USA, August 2008.

[107] Abderrahmen Mtibaa, Martin May, and Mostafa Ammar. On the relevance of social information to opportunistic forwarding. *Modeling, Analysis, and Simulation of Computer Systems, International Symposium on*, 0:141–150, August 2010.

[108] Abderrahmen Mtibaa, Martin May, Christophe Diot, and Mostafa Ammar. PeopleRank: Social opportunistic forwarding. In *IEEE INFOCOM 2010*. IEEE, March 2010.

[109] Derek G. Murray, Eiko Yoneki, Jon Crowcroft, and Steven Hand. The case for crowd computing. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 39–44, New York, NY, USA, 2010. ACM.

[110] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[111] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[112] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[113] Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, December 1987.

[114] Antonis Panagakis, Athanasios Vaios, and Loannis Stavrakakis. On the effects of cooperation in DTNs. In *Proceedings of the 2nd International Conference on Communication Systems Software and Middleware, 2007. COMSWARE 2007*, pages 1–6. IEEE, January 2007.

[115] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, November 2006.

[116] Charles Perkins and Ian Chakeres. Dynamic MANET on-demand (DYMO) routing. Technical Report draft-ietf-manet-dymo-21.txt, IETF Secretariat, Fremont, CA, USA, July 2010.

[117] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-Demand distance vector routing. *Second IEEE Workshop on Mobile Computer Systems and Applications*, 0:90, 1999.

[118] Andreea Picu and Thrasyvoulos Spyropoulos. Distributed stochastic optimization in opportunistic networks: the case of optimal relay selection. In *Proceedings of the 5th ACM workshop on Challenged networks*, CHANTS '10, pages 21–28, New York, NY, USA, 2010. ACM.

[119] Anna-Kaisa Pietiläinen, Earl Oliver, Jason Lebrun, George Varghese, and Christophe Diot. MobiClique: Middleware for mobile social networking. In *Proceedings of the Second ACM SIGCOMM Workshop on Online Social Networks*, August 2009.

[120] Calicrates Policroniades, Pablo Vidales, Martin Roth, and Daniel Kreienbühl. Data management in human networks. In *CHANTS '07: Proceedings of the second workshop on Challenged networks CHANTS*, pages 83–90, New York, NY, USA, 2007. ACM.

[121] J. M. Pujol, A. Lopez Toledo, and P. Rodriguez. Fair Routing in Delay Tolerant Networks. In *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pages 837–845. IEEE, April 2009.

[122] Giovanni Resta and Paolo Santi. The effects of node cooperation level on routing performance in delay tolerant networks. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON'09, pages 413–421, Piscataway, NJ, USA, 2009. IEEE Press.

[123] L. Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 1(1):73–90, 1979.

[124] James Scott, Jon Crowcroft, Pan Hui, and Christophe Diot. Haggle: a networking architecture designed around mobile users. In *Proceedings of the Third Annual Conference on Wireless On-demand Network Systems and Services*, 2006.

[125] U. Shevade, Han H. Song, Lili Qiu, and Yin Zhang. Incentive-aware routing in DTNs. In *2008 IEEE International Conference on Network Protocols*, pages 238–247. IEEE, October 2008.

[126] John Solis, N. Asokan, Kari Kostiainen, Philip Ginzboorg, and Jörg Ott. Controlling resource hogs in mobile delay-tolerant networks. *Computer Communications*, 33(1):2–10, January 2010.

[127] Libo Song and David F. Kotz. Evaluating opportunistic routing protocols with large realistic contact traces. In *CHANTS '07: Proceedings of the second workshop on Challenged networks CHANTS*, pages 35–42, New York, NY, USA, 2007. ACM.

[128] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi 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, pages 252–259, New York, NY, USA, 2005. ACM.

[129] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Efficient routing in intermittently connected mobile networks: The Multiple-Copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90, February 2008.

[130] Vikram Srinivasan, Mehul Motani, and Wei T. Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 86–97, New York, NY, USA, 2006. ACM.

[131] Vikram Srinivasan, Mehul Motani, and Wei T. Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *MobiCom '06*, pages 86–97, New York, NY, USA, 2006. ACM.

[132] Vikram Srinivasan, Mehul Motani, and Wei T. Ooi. CRAWDAD data set nus/contact (v. 2006-08-01). Downloaded from http://crawdad.org/nus/contact, August 2006.

[133] J. Stiller and R. Dunbar. Perspective-taking and memory capacity predict social network size. *Social Networks*, 29(1):93–104, January 2007.

[134] Jing Su, Kelvin K. W. Chan, Andrew G. Miklas, Kenneth Po, Ali Akhavan, Stefan Saroiu, Eyal de Lara, and Ashvin Goel. A preliminary investigation of worm infections in a bluetooth environment. In *WORM '06: Proceedings of the 4th ACM workshop on Recurring malcode*, pages 9–16, New York, NY, USA, 2006. ACM.

[135] Yan L. Sun, Zhu Han, Wei Yu, and K. J. Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *IEEE INFOCOM*, pages 230–236, 2006.

[136] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226, New York, NY, USA, 2004. ACM Press.

[137] Gautam S. Thakur, Ahmed Helmy, and Wei J. Hsu. Similarity analysis and modeling in mobile societies: the missing link. In *Proceedings of the 5th ACM workshop on Challenged networks*, CHANTS '10, pages 13–20, New York, NY, USA, 2010. ACM.

[138] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[139] Md Y. Uddin, Brighten Godfrey, and Tarek Abdelzaher. RELICS: In-network realization of incentives to combat selfishness in DTNs. In *Proceeding of the 18th IEEE International Conference on Network Protocols (ICNP)*, pages 203–212, October 2010.

[140] Brandes Ulrik and Thomas Erlebach, editors. *Network Analysis: Methodological Foundations (Lecture Notes in Computer Science / Theoretical Computer Science and General Issues)*. Springer, first edition, March 2005.

[141] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks, 2000.

[142] Amin Vahdat and David Becker. Epidemic routing for Partially-Connected ad hoc networks. Technical report, Duke University, April 2000.

[143] Stanley Wasserman. *Social network analysis : methods and applications*. Cambridge [England] ; New York : Cambridge University; Press, 1997., 1997.

[144] Douglas R. White and Karl P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2), 1983.

[145] Harrison C. White, Scott A. Boorman, and Ronald L. Breiger. Social structure from multiple networks. i. blockmodels of roles and positions. *The American Journal of Sociology*, 81(4):730–780, 1976.

[146] Kuang Xu, Pan Hui, Victor, Jon Crowcroft, Vito Latora, and Pietro Lio. Impact of altruism on opportunistic communications. In *Proceedings of the first international conference on Ubiquitous and future networks*, ICUFN'09, pages 153–158, Piscataway, NJ, USA, 2009. IEEE Press.

[147] Kuang Xu, Victo Li, and Jaewoo Chung. Exploring centrality for message forwarding in opportunistic networks. In *IEEE WCNC 2010*, pages 1–6. IEEE, April 2010.

[148] Ting Yan, Tian He, and John A. Stankovic. Differentiated surveillance for sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 51–62, New York, NY, USA, 2003. ACM Press.

[149] Jihwang Yeo, David Kotz, and Tristan Henderson. CRAWDAD: a community resource for archiving wireless data at dartmouth. *Computer Communication Review*, pages 21–22, 2006.

[150] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 225–234, New York, NY, USA, 2007. ACM.

[151] Eiko Yoneki, Pan Hui, and Jon Crowcroft. Visualizing community detection in opportunistic networks. In *Proceedings of the second ACM workshop on Challenged networks*, CHANTS '07, pages 93–96, New York, NY, USA, 2007. ACM.

[152] Bin Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 1–10, December 2004.

[153] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, 2006.

[154] Qiang Zhou, Jing Ying, and Minghui Wu. A detection method for uncooperative nodes in opportunistic networks. In *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2010)*, pages 835–838. IEEE, September 2010.

[155] Haojin Zhu, Xiaodong Lin, Rongxing Lu, Yanfei Fan, and Xuemin Shen. SMART: A secure multilayer Credit-Based incentive scheme for Delay-Tolerant networks. *IEEE Transactions on Vehicular Technology*, 58(8):4628–4639, October 2009.