# NEIGHBOUR DISCOVERY AND DISTRIBUTED SPATIO-TEMPORAL CLUSTER DETECTION IN POCKET SWITCHED NETWORKS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2013

By
Matthew Orlinski
School of Computer Science

# Contents

Word Count: 35927

# List of Tables

# List of Figures

# Abstract


Neighbour discovery and distributed spatio-temporal cluster detection in pocket switched networks

Matthew Orlinski

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2013

Pocket Switched Networks (PSNs) offer a means of infrastructureless inter-human communication by utilising Delay and Disruption Tolerant Networking (DTN) technology. However, creating PSNs involves solving challenges which were not encountered in the Deep Space Internet for which DTN technology was originally intended.

End-to-end communication over multiple hops in PSNs is a product of short range opportunistic wireless communication between personal mobile wireless devices carried by humans. Opportunistic data delivery in PSNs is far less predictable than in the Deep Space Internet because human movement patterns are harder to predict than the orbital motion of satellites. Furthermore, PSNs require some scheme for efficient neighbour discovery in order to save energy and because mobile devices in PSNs may be unaware of when their next encounter will take place.

This thesis offers novel solutions for *neighbour discovery* and *opportunistic data delivery* in PSNs that make practical use of dynamic inter-human encounter patterns.

The first contribution is a novel neighbour discovery algorithm for PSNs called PISTONS which relies on a new inter-probe time calculation (IPC) and the bursty encounter patterns of humans to set the time between neighbour discovery scans. The IPC equations and PISTONS also give participants the ability to easily specify their required level of connectivity and energy saving with a single variable.

This thesis also contains novel *distributed spatio-temporal clustering* and opportunistic data delivery algorithms for PSNs which can be used to deliver data over multiple hops. The spatio-temporal clustering algorimths are also used to analyse the social networks and transient groups which are formed when humans interact.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright

# Acknowledgements

I would like to thank and acknowledge the support of Dr. Nick Filer who gave me the confidence and freedom to explore the topics in this thesis.

I would also like to extend my sincerest gratitude to Dr. Barry Cheetham and the entire APT group. They have been a constant source of help, encouragement, and fun throughout my time in Manchester.

Thank you also to Dr Michael Sheldon for proofreading this document on his boat whilst coming up with wonderful (terrible) zebra jokes, and my viva committee of Professor Madjid Merabti and Dr Richard Banach.

Thank you also to Ari Keränen and the ONE simulator team, Pan Hui for his work on distributed cluster detection, Jon Crowcroft for his encouragement at Cosener's, and PJ Dillon for his implementation of Bubble.

Finally, I'd like to thank Chris, Jean, and Sam for being there when I needed it. Their unconditional support made this possible, and I am forever in their debt.

# Acronyms

| | |
|---|---|
| **AODV** | Ad hoc On-Demand distance Vector routing |
| **BBR** | Bluetooth 4.0: Basic Rate |
| **BLE** | Bluetooth 4.0: Low Energy |
| **CAR** | Context-aware Adaptive Routing |
| **CRAWDAD** | Community Resource for Archiving Wireless Data At Dartmouth |
| **DEBT** | Distributed Expectation-Based spatio-Temporal clustering |
| **DEBTC** | Distributed Expectation-Based spatio-Temporal Cluster based message delivery |
| **DEBTE** | Distributed Expectation-Based spatio-Temporal Epidemic based message delivery |
| **DEBTT** | Distributed Expectation-Based spatio-Temporal Tree based message delivery |
| **DOSN** | Distributed Online Social Network |
| **DPL** | Densification Power Law |
| **DRAFT** | Distributed Rise And Fall spatio-Temporal clustering |
| **DTN** | Disruption or Delay Tolerant Network |
| **DWARF** | DTN-oriented Wireless interface Activation mechanism based on Radio Fluctuations |
| **IoT** | Internet of Things |
| **IPC** | Inter-Probe time Calculation |
| **MANET** | Mobile Ad hoc NETwork |
| **MEBS** | Multi frame Expectation-Based Spatio-temporal clusters |
| **N4C** | Networking for Communications Challenged Communities |

| | |
|---|---|
| **NDREP** | Neighbour Discovery Reply |
| **NDREQ** | Neighbour Discovery Request |
| **NFC** | Near Field Communication |
| **OLSR** | Optimised Link State Routing |
| **PCC** | Pearson Correlation Coefficient |
| **PMWD** | Personal Mobile Wireless Device |
| **PRoPHETv2** | Probabilistic ROuting Protocol using History of Encounters and Transitivity version 2 |
| **PSN** | Pocket Switched Network |
| **RWM** | Random Walk Movement model |
| **SEBS** | Single frame Expectation-Based Spatio-temporal clusters |
| **STAR** | Short Term Arrival Rate |
| **SSNDI** | Synchronised Symmetric Neighbour Discovery Interval |
| **The ONE** | The Opportunistic Network Environment simulator |
| **VANET** | Vehicular Ad hoc Network |
| **WDM** | Working Day Movement model |
| **Wi-Fi** | IEEE 802.11 - 2007 revision |

# Chapter 1

# Introduction

Over 1.8 billion mobile phones that communicate with each other via cellular networks will be sold worldwide in 2013 [Milanesi et al., 2013]. Despite this huge number of devices, many people in the world's developing and remote regions will have no access to cellular communications. Figure 1.1 shows the extent of the population unable to communicate via cellular technology in some Sub-Saharan African countries, but there are many more places in Northern Europe and beyond that share this problem [Grasic et al., 2011]. This *coverage problem* is partly due to the prohibitive cost of building and maintaining the cell towers which connect mobile phones to the network operator's core network, as well as the cost of spectrum licences required to operate the cell towers [Deloitte and GSMA, 2012].

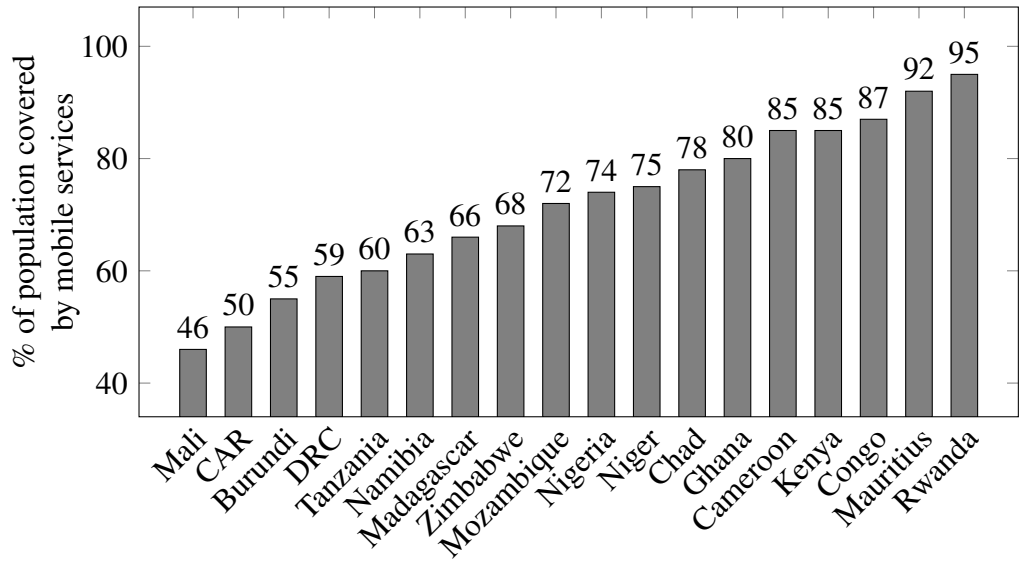



Figure 1.1: Percentage of the population covered by cellular networks in some African countries as of November 2012 [Deloitte and GSMA, 2012].

As well as increasing cellular coverage for people in developing and remote regions, cellular network operators also need satisfy increasing end-user demand for real time data streaming in developed countries. In what is sometimes called the *capacity problem* [Balasubramanian et al., 2010a], operators are under constant pressure to increase the capacity of their networks as the amount of mobile data traffic rises at a compound annual growth rate of 78 percent and is expected to reach 10.8 exabytes per month by 2016 [Cisco, 2012].

In order to cope with both the coverage and capacity problems the communication range of cells has been getting smaller [López-Pérez et al., 2009]. Smaller cells such as those shown in Figure 1.2 allow cellular network operators to optimise spectrum use, increase network capacity and coverage, and reduce operating costs [Andrews et al., 2012].

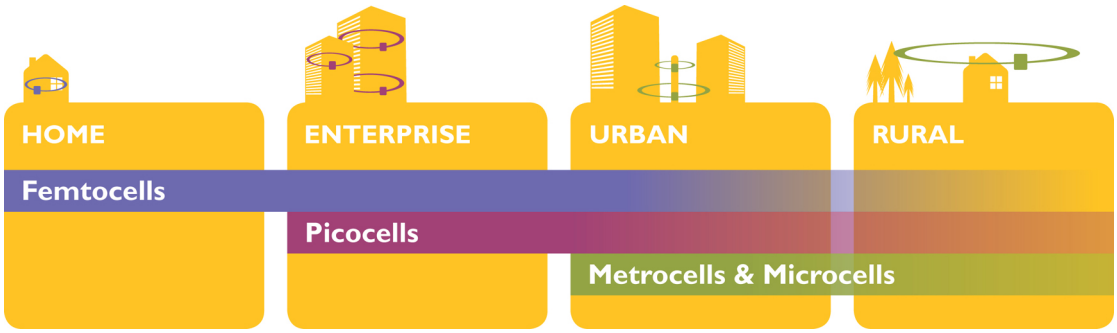



Figure 1.2: The continuum of small cell technology. This image used with permission from the Small Cell Forum [Mansfield and Wright, 2013].

However, small cells come with their own challenges. Femto and picocells can interfere with metrocells [López-Pérez et al., 2009], and smaller cells still need to be connected to the cellular operator's core network, which may be expensive or challenging in rural areas.

## 1.1 Motivation

This thesis looks at an orthogonal solution to the coverage problem, one that does not require the installation of more metrocells or connections to smaller cells in remote regions of the world. This work explores a method for inter-human data exchange where singular blocks of data referred to as *messages* (or bundles [Scott, 2007]) need not travel via a cellular network. Instead, messages are exchanged *opportunistically* between Personal Mobile Wireless Devices (PMWDs) carried by humans using short range wireless connections [Han et al., 2012a]. Other than the cost of the devices, opportunistic networking is free as it can use unlicensed spectrum such as the 2.4 GHz frequency band currently being used by IEEE 802.11 Wi-Fi [Kerry, 2007].

The term PMWD as used in this thesis is a general term referring to *homogeneous* personal wireless devices that can communicate with each other using opportunistic ad hoc data connections whilst moving. The class of devices that PMWDs refers includes mobile phones, but may also include other devices carried by humans such as laptops, watches, and anything else capable of digital wireless communication. This thesis does not discuss issues involving portable wireless devices such as laptops which are movable but that typically only communicate whilst stationary.

## 1.2 Opportunistic networking

Opportunistic message delivery protocols [Hui et al., 2006, Grasic et al., 2011] utilise the mobility of participants and the occasional connections between them to propagate messages toward their intended destination. As digital memory is cheap and portable, PMWDs can act as mobile caches for messages which are intended for other PMWDs.

Opportunistic networks created using PMWDs are commonly referred to as Pocket Switched Networks (PSNs) because the networks are created using PMWDs that are often carried in participants' pockets [Chaintreau et al., 2005, Hui et al., 2006]. PSN research usually assumes that communication range between nearby PMWDs will be in the region of 10 to 30 meters to be compatible with the Bluetooth [SIG, 2010] and IEEE 802.11 Wi-Fi [Kerry, 2007] specifications as implemented on modern mobile phones.

Inter-device communication in PSNs is challenging because of the short communication range, and potentially large distance between participants. Frequent disconnections between participants due to human movement patterns mean that PSNs may not be suitable for all real-time applications such as voice and video streaming. However, PSNs offer a solution to coverage and capacity problems where it would otherwise be too expensive or difficult to do otherwise. Continued PSN research is expected to count toward a number of long term goals:

1. Free communication between PMWDs using infrastructureless communications with fully distributed coordination.

2. Continued, albeit limited, communication during unplanned outages of cellular communication infrastructure.

3. Basic communication for developing regions of the world that can be used to retrieve information from sensors, send and receive emails, etc.

4. Provide some relief for overloaded cellular networks.

In order to contribute to the realisation of PSNs, this thesis looks at the challenges experienced when deploying so called Delay or Disruption Tolerant Networks (DTNs) [Liu and Wu, 2007, Lindgren and Hui, 2009, Balasubramanian et al., 2010b]. The justification for why disruption tolerant networking is suitable for PSNs is given over the next three sections, starting with a description of more general mobile ad hoc networks.

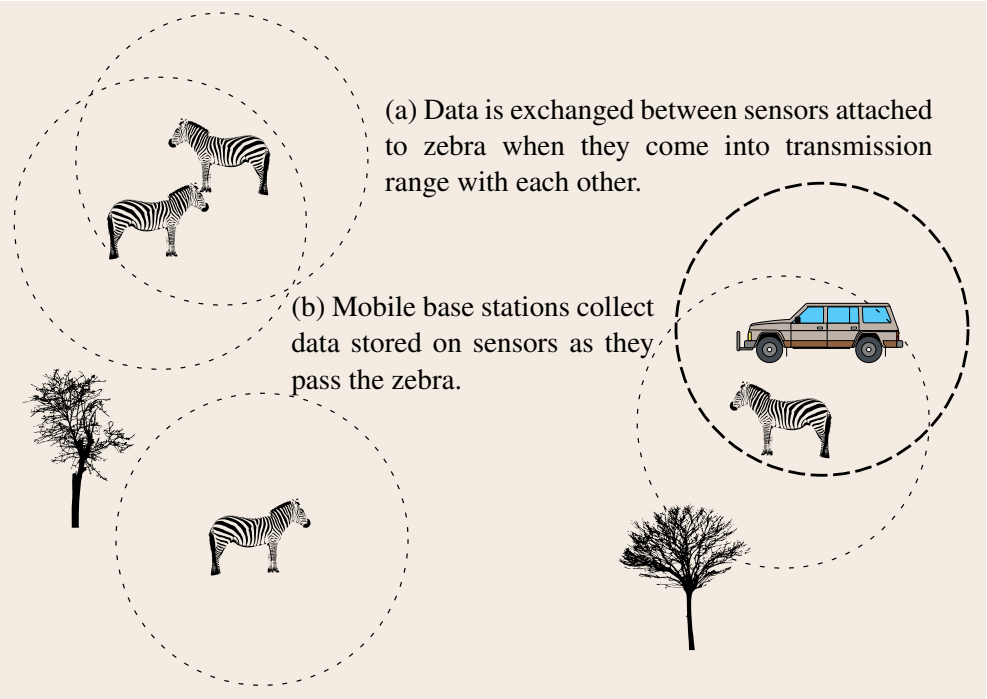



Figure 1.3: An example of the operation of ZebraNet. The mobile base station does not need to encounter all of the zebra in order to gather the information it needs because sensors attached to zebra exchange information with each other.

## 1.3 Mobile ad hoc networks

Mobile Ad hoc Networks (MANETs) are created when mobile electronic devices form ad hoc, peer-to-peer, connections between themselves and/or stationary devices using wireless connections. MANETs have been a popular research topic and have continued to evolve since the 1990s [Royer and Toh, 1999]. There is no exhaustive list of applications for MANETs, nor is there one specific configuration of devices and mobility which is prevalent in the field of MANET research.

In recent years MANETs have been used for the collection of sensor network data. For example, the MoDisNet system [Ma et al., 2008] is used to monitor air pollution and combines a high-throughput static sensor grid with mobile sensors mounted on cars. MANETs created using a mixture of static and mobile sensors attached to vehicles are often referred to as Vehicular Ad hoc Networks (VANETs) [Fiore et al., 2007]. The CarTel project [Hull et al., 2006] also uses a VANET to collect and process sensor data. The data collected by CarTel sensors is transmitted to base stations opportunistically during rare periods of wireless connectivity with static base stations, or by utilising *data mules* [Shah et al., 2003] such as a driver's PMWDs.

The ZebraNet wildlife tracker [Juang et al., 2002] uses low power, short range sensor devices to collect detailed movement traces of zebra at the Sweetwaters Game Reserve near Nanyuki, Kenya. As zebra are free to roam around the reserve and often

form tight-knit harems, collars containing the sensor devices attached to zebra cannot assume connectivity to a base station or even other sensors. Instead of discarding data which cannot be transmitted right away, ZebraNet sensors store data and use periodic neighbour discovery to look for opportunities where the data can be transferred to base stations or other sensors as shown in Figure 1.3.

The *store-and-forward* communication paradigm adopted by CarTel and ZebraNet, enables data to propagate in the direction of base stations for processing even when there are no stable end-to-end connections between the source and destination. The store-and-forward paradigm also forms the basis of all communication in Disruption or Delay Tolerant Networks (DTNs) [Small and Haas, 2005, Liu and Wu, 2007, Lindgren and Hui, 2009, Balasubramanian et al., 2010b].

## 1.4 Delay or disruption tolerant networks

Successful adoption of the store-and-forward paradigm ensures that data is not lost during periods of time when there are no stable end-to-end paths between the source and destination. Instead of dropping data which cannot be transmitted immediately, devices store data until it can be transmitted, thus ensuring that data can gradually work its way toward its destination.

The major problems encountered by DTNs were described succinctly during the creation of the Deep Space Internet [Brown et al., 2008] with which information from Mars has recently been relayed back to Earth using the Epoxi spacecraft [Hooke, 2010]. Experiments into the provision of the Deep Space Internet by the National Aeronautics and Space Administration (NASA) highlighted two major challenges for all DTN protocols [Scott, 2007]:

1. **Delay.** Devices may be very far apart, especially in the Deep Space Internet where the speed of light constrains communication over very large distances.

2. **Disruption.** Communication between devices may be disrupted as a consequence of movement or environmental changes. To continue the Deep Space Internet example, orbital motion can disrupt communication between satellites or base stations on Earth when there is no line of sight for radio communications.

As well as gathering data from space, DTNs can facilitate data delivery between participants on Earth when there are frequent disruptions to wireless communication. For example, participatory sensing applications [Reddy et al., 2011] (participatory sensing is the process whereby human communities use PMWDs to collect data important to them) use DTN technology to opportunistically transfer data collected by PMWDs to static processing stations or to each other in a similar way as data is exchanged in ZebraNet. The next section describes some of the other possibilities, as well as the limitations, of applying DTN technology to inter-human communication.

## 1.5 Pocket switched networks

Real-time inter-human communication protocols such as VoIP require a maximum 150 millisecond one-way latency and high packet delivery success rates in order to give the effect of free flowing conversation [It, 1993]. Real-time communication in PSNs is currently difficult due to the movement patterns of humans and short transmission range of PMWDs. So what is the point of PSN research? And how can PSNs aide inter-human communication in the future? These questions are briefly addressed in this section.

The Isis wireless payment system [Contini et al., 2011], a joint venture by major telecommunications companies AT&T, Verizon Wireless, and T-Mobile, was launched in the United States in September 2012. Isis relies on Near Field Communication (NFC) [Ross, 2012], and works by users moving their PMWDs very close (typically under 10 centimetres) to payment stations in order to make a payment. Isis can also be used to complete payments between PMWDs. For example, a waiter in a restaurant can carry his own PMWD in order to complete NFC transactions. The waiter's PMWD can then opportunistically transfer completed transactions to a base station using short range wireless signals at some later point in time. In this case the waiter becomes the data mule (an unfortunate expression in this context), who acts as a DTN node providing a link between the customer's PMWD and the payment service provided by the credit card company.

PMWDs can act as gateways to external services using DTN technology, but the goal of PSNs such as PeopleNet [Motani and Srinivasan, 2005] is primarily to facilitate inter-human peer-to-peer communication. The challenges faced by PSNs are similar to those faced in the Deep Space Internet. However, human movement patterns are harder to predict than the orbital motion of satellites.

The Networking for Communications Challenged Communities (N4C) project has successful demonstrated that it is possible to use PMWDs moving between populated areas and static Internet gateways to solve the coverage problem in the Kočevje region in Slovenia and in Lapland [Grasic et al., 2011]. Communication using this access method is slow, with end-to-end latency depending on the movement speeds and patterns of people between Internet gateways and end receivers. However, PSN research has provided a means of communication which satisfies many of the needs of these communities (e.g. email and social networking updates) where it would be too expensive to do otherwise

As well as PSNs, there are many other so called "killer applications" being proposed elsewhere for DTN technology. For example, communications in the presence of oppressive governments [Lindgren and Hui, 2009], and a free alternative to cellular communication in urban areas. PSNs and the resulting *internet of people* may sound dystopian to some, and have some unpleasant implications if privacy is not handled correctly, but such a system could also save lives. For example, a PMWD in a PSN could alert others nearby that their owner is in danger if the objectives outlined in the next section are met [Huang et al., 2005].

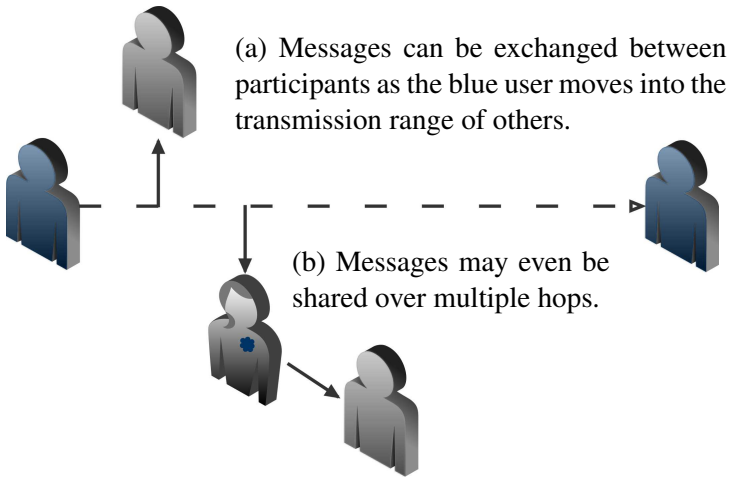



Figure 1.4: PSNs can be used to pass messages between participants.

## 1.6 Objectives

Figure 1.4a depicts a user carrying a PMWD moving into transmission range of other users with their own PMWD. If this scenario were happening in an enclosed area such as a conference or office space, or an area where there is no cellular coverage, then it may be preferable for PMWDs to exchange information such as business cards or internal memos opportunistically. The data transferred can even travel over multiple hops between participants who never come into direct communication range with each other, as is depicted in Figure 1.4b.

If autonomous wireless networking between PMWDs was more prevalent, then the scene described in Figure 1.4 where messages are being opportunistically passed between PMWDs may be an every day occurrence. However, before the scenario described in Figure 1.4 can be fully realised there are some outstanding issues which need to be addressed. This thesis looks at how PMWDs can reliably detect the presence of other PMWDs in transmission range (an aspect of self-configuration known as neighbour discovery [Vasudevan et al., 2005]), and how messages can be exchanged between PMWDs assuming the address of the destination PMWD is known.

### 1.6.1 Neighbour discovery

Caching of known neighbours implies an element of exploitable stability within the network which is not the case in PSNs where rapid topological changes occur as a consequence of dynamic human mobility patterns [Birrane III et al., 2011]. It is therefore critically important that neighbour discovery procedures for PSNs are autonomous and can be completed without the benefit of a controlling cell or access point. However, neighbour discovery in PSNs must also be as efficient as possible because of the following limitations:

1. Battery power is one of the most fundamental constraints for PMWDs. Moreover, neighbour discovery tends to be more expensive in terms of energy usage than maintaining existing connections [Perrucci et al., 2011].

2. PMWDs cannot be in a constant state of neighbour discovery as this would waste the bandwidth needed to satisfy Quality of Service (QoS) commitments.

Chapter 3 assesses the suitability of neighbour discovery in Bluetooth for PSNs. Chapter 3 also states the case for using symmetric neighbour discovery intervals for autonomous neighbour discovery in PSNs, which is also the same method which was used for neighbour discovery in ZebraNet [Liu et al., 2004].

### 1.6.2 Opportunistic message delivery using clusters

The second objective of this research is to provide PSNs with a reliable opportunistic message delivery protocol that does not rely on cellular infrastructure or some other centralised hierarchy.

Multi-copy opportunistic messages delivery protocols such as Epidemic specify that messages should be copied to every encountered PMWD until the destination is encountered [Matsuda and Takine, 2008]. Flooding the network in this way can increase the likelihood that a message will be delivered when the movement of participants is unpredictable, but it can also create many duplicate messages which waste the energy, bandwidth, and storage capacity of PMWDs.

Targeted messaged delivery using cluster labels such as those shown in Figure 1.5 has been shown to limit the size of the number of duplicate messages needed to reach the destination, even when the probability of a device being able to deliver a message is unknown or difficult to calculate [Hui and Crowcroft, 2007]. Furthermore, by clustering PMWDs which have similar encounter patterns using distributed cluster detection algorithms [Hui et al., 2007a], PMWDs have a decentralised mechanism with which to partition the PSN and limit message duplication.

With the help of Figure 1.5 an example is now given which will clarify how cluster based message delivery can increase message delivery efficiency in PSNs. Imagine that a PMWD called $v_i$ wishes to send a message to another called $v_k$, but $v_i$ does not know the exact location or have the ability to find the quickest path to $v_k$. Now also imagine that a PMWD's cluster label can be sent to other PMWDs in transmission range. Then one process with which $v_i$ might get a message to $v_k$ can be summarised as follows:

**Stage 1.** $v_i$ wants to send a message to $v_k$. $v_i$ comes into transmission range of $v_j$ which is in cluster $B$. $v_j$ also reports that Cluster B contains $v_k$ (the message destination), so $v_i$ sends a copy of the message to $v_j$. $v_i$ does not need to duplicate this message further unless it meets another PMWD from cluster B.

**Stage 2.** $v_j$ is not currently connected to $v_k$, but $v_j$ knows that it belongs to the same cluster as $v_k$ and can copy the message to whoever it encounters within the cluster boundary.

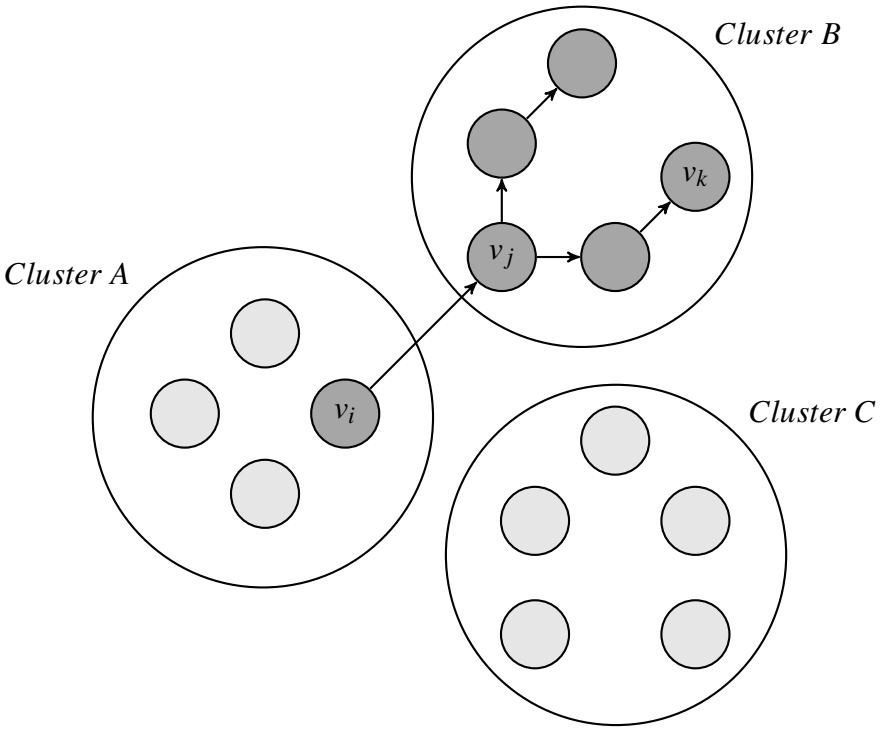



Figure 1.5: Opportunistic message delivery example between PMWDs $v_i$ and $v_k$ using cluster boundaries to limit message duplication.

**Stage 3.** Stage 2 is repeated by other PMWDs in Cluster B which have the message until the message is delivered to $v_k$.

Upon consideration of this simplistic example based on the scenario in Figure 1.5, it may be apparent that the number of, size, and membership of the clusters will impact upon message duplication. In a more detailed description of the opportunistic message delivery problem in PSNs there is also the added complication that clusters must be generated by PMWDs themselves using distributed cluster detection algorithms, and confirmation that a message has been delivered cannot be sent across a wide area.

From Chapter 4 onwards, this thesis looks at opportunistic message delivery protocols which utilise clusters, and novel approaches for both distributed cluster detection and opportunistic message delivery for PSNs are proposed.

## 1.7 Methodology

The PSN experiments described in this thesis make use of either existing reality mining datasets [Eagle and (Sandy) Pentland, 2006] or synthetic movement models in order to generate simulated encounters between participants. An introduction to both reality mining and synthetic movement models is given in this section.

| | Infocom5 | Infocom6 | Cambridge | UCL1 | Reality |
|---|---|---|---|---|---|
| **Environment** | Conference | | Campus | | |
| **Duration (days)** | 3 | 3 | 12 | 6 | 246 |
| **Number of PMWDs** | 41 | 78 | 36 | 20 | 97 |
| **Device type** | iMote | iMote | iMote | Phone | Phone |
| **Number of encounters** | 22459 | 128979 | 10641 | 512 | 102594 |
| **Daily encounter probability** | 0.78 | 0.73 | 0.24 | 0.53 | 0.01 |
| **Granularity (seconds)** | 120 | 120 | 600 | n/a | 300 |
| **Geographic location** | No | No | No | GPS | Cell ID |

Table 1.1: Comparison of some reality mining datasets. Encounters between participants are logged using Bluetooth apart from in UCL1 where encounters are inferred from GPS and Wi-Fi logs. The daily encounter probability is the probability that an encounter with a particular other PMWD will take place on any given day, and the granularity is the time between neighbour discovery processes.

### 1.7.1 Reality mining datasets

Reality mining experiments typically collect human behavioural data using PMWDs belonging or given to participants to use in an unsupervised manner as they go about their daily lives. The information collected can include but is not limited to:

1. The results of neighbour discovery scans using Bluetooth and Wi-Fi.

2. Call and text message logs.

3. Information from sensors such as microphones, accelerometers, and the Global Positioning System (GPS).

PMWDs equipped with Bluetooth and Wi-Fi can be used to scan their surroundings for other nearby PMWDs. The data collected can then be used to analyse when and for how long participants come into close proximity with one another. Each of the reality mining datasets in Table 1.1 is available from the CRAWDAD repository,[1] and each can be used by the Opportunistic Network Environment (ONE) simulator [Keranen et al., 2009] to recreate the encounters between PMWDs for opportunistic message delivery experiments.

[1] Many more reality mining datasets are available at http://crawdad.cs.dartmouth.edu/.

1. The Infocom5, Infocom6, and Cambridge datasets were collected as part of the Haggle project from the University of Cambridge [Scott et al., 2009]. Haggle was also one of the first projects to demonstrate that opportunistic wireless connections between PMWDs can be used to deliver messages between participants.

2. The Reality dataset (Note: Not to be confused with reality mining) was compiled using one hundred mobile phones logging encounters with Bluetooth devices on the MIT campus over a period of nine months [Eagle and Pentland, 2005].

3. UCL1 is the first of the two privacy studies conducted at University College London in conjunction with the University of St Andrews [Abdesslem et al., 2011]. Encounters between participants in this dataset were inferred if GPS information indicated that participants were within 10 metres of one another.

The figures given in Table 1.1 for numbers of PMWDs and encounters between them only take into account the mobile devices of participants. Any interactions with static devices or devices external to the experiment are omitted from all analysis in this thesis. This was for two reasons; Firstly, it was unclear which external devices were static and which were mobile. Secondly, some of the external devices found were only seen once, and therefore would not make a significant contribution to opportunistic message delivery experiments.

### 1.7.2 Synthetic movement models

The number of reality mining datasets is finite, where more encounter data is required a number of synthetic movement models can be used to generate encounters between virtual participants.

The accuracy of synthetic models for human movement and encounter simulations has been tested against reality mining data. For instance, typical inter-encounter time distributions for human movement patterns have been calculated from reality mining data and are accurately replicated in the Working Day Movement (WDM) model [Ekman et al., 2008]. In contrast, random movement models [Yoon et al., 2003] do not accurately reflect human mobility and should therefore be used with caution in PSN simulations [Chaintreau et al., 2005].

Synthetic movement models are more suited to neighbour discovery experiments than reality mining datasets because it is not easy to retrospectively alter the granularity with which reality mining data was collected. Chapter 3 explores neighbour discovery for virtual PSNs using the WDM model, but it also includes results using a random movement model to show the effects of both models on simulations.

## 1.8 Contributions

This thesis makes the following contributions by exploring the objectives from Section 1.6 using the methodologies described in Section 1.7:

1. Chapter 3 explores the effects of altering the time between *symmetric neighbour discovery* processes on connectivity in PSNs. This work also resulted in a new neighbour discovery algorithm for PSNs called PISTONS.

2. Two new distributed cluster detection algorithms for PSNs called Promote-2 and Nomads are introduced in Chapter 4. This work also contains a detailed description of the clusters produced by existing distributed cluster detection algorithms.

3. Following the observation that human movement and encounter patterns change over time [Hui, 2008, Zyba et al., 2011], Chapter 5 introduces two *expectation-based spatio-temporal clustering* algorithms for dynamic encounter graphs that can be used to analyse reality mining data and detect when humans group together. This work involved:

   (a) Analysing reality mining data in consecutive discrete time frames to show that inter-human interactions differ at different times.

   (b) Calculation of the rate at which new edges are added to graphs used to representing reality mining datasets.

   (c) Two new expectation-based spatio-temporal clustering algorithms that detect new spatio-temporal clusters when human encounter patterns change.

4. Two novel distributed spatio-temporal cluster detection algorithms. A distributed expectation-based spatio-temporal cluster detection algorithm called DEBT is proposed in Chapter 6, and a *non-monotonic* distributed spatio-temporal cluster detection algorithm called DRAFT is introduced in Chapter 7 (the meaning of the term monotonic in regards to clustering is described at the start of Chapter 5).

5. New opportunistic message delivery algorithms that improve upon the message delivery probability and efficiency of current state of the art algorithms for DTNs and PSNs:

   (a) The tree based opportunistic message delivery algorithm proposed in Chapter 6 for use with DEBT utilises spatio-temporal clusters to deliver more messages, more efficiently, than existing opportunistic message delivery algorithms that also use clusters in forwarding decisions.

   (b) Chapter 7 shows that opportunistic message delivery using DRAFT clusters can deliver more messages successfully than some existing algorithms for DTNs and PSNs which do not use clustering.

## 1.9 List of publications

The following four papers have been published in support of this work:

1. Orlinski, M. and Filer, N. (2013) 'The rise and fall of spatio-temporal clusters in mobile ad-hoc networks', Journal of Ad Hoc Networks, vol. 11, no. 5.

2. Orlinski, M. and Filer, N. (2012) 'Distributed expectation-based spatio-temporal cluster detection for pocket switched networks', Proceedings of the IFIP Wireless Days Conference, Dublin, Ireland.

3. Orlinski, M. and Filer, N. (2012) 'Movement speed based inter-probe times for ad-hoc wireless devices', Proceedings of the Fourth International Conference on Ad Hoc Networks, Paris, France.

4. Orlinski, M. and Filer, N. (2012) 'Quality distributed community formation for data delivery in pocket switched networks', Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners, Lyon, France.

## 1.10 Thesis structure

The remainder of this thesis is structured as follows:

**Chapter 2.** Acts as the literature review and introduces the data analysis techniques which are used throughout this thesis.

**Chapter 3.** Contains the contributions relating to efficient neighbour discovery for PSNs, including the new PISTONS algorithm.

**Chapter 4.** Explores existing distributed cluster detection algorithms that can be used in PSNs. New terminology is introduced with which to describe the clusters produced by the algorithms, and both the positive and negative effects of using clusters in opportunistic forwarding decisions are discussed.

**Chapter 5.** Introduces novel expectation-based spatio-temporal cluster detection algorithms for dynamic encounter graphs.

**Chapter 6.** Proposes ways of implementing expectation-based spatio-temporal cluster detection algorithms in a distributed system such as a PSN.

**Chapter 7.** Uses the terminology from chapters 4, 5, and 6 to describe a new non-monotonic distributed cluster detection and opportunistic message delivery algorithm for PSNs called DRAFT.

**Chapter 8.** Concludes the discussion and offers some possible avenues of further exploration.

# Chapter 2

# Related work

This chapter contains subsidiary information related to the topics and analysis techniques used throughout this thesis. It begins with a discussion on how Personal Mobile Wireless Devices (PMWDs) in a distributed and decentralised system can detect *opportunistic* encounters with each other, and how PMWDs can exchange messages over transient wireless connections. This chapter then goes on to talk about how clusters of PMWDs with similar encounter patterns are formed, how these clusters are analysed, and how clusters might be used to increase message delivery efficiency in PSNs.

## 2.1 Neighbour discovery for PSNs

In order that PMWDs belonging to different individuals can discover and communicate with each other, each PMWD must use the same neighbour discovery procedure which defines how to transmit signals to and receive signals from other nearby PMWDs. This section will discuss some of the currently available solutions for, and on-going research into neighbour discovery for PSNs.

Generally, the different signals sent between PMWDs during neighbour discovery procedures can be categorised as either Neighbour Discovery Requests (NDREQs) and Neighbour Discovery Replies (NDREPs) [Hui, 2008, Izumikawa et al., 2010], or independent neighbour discovery beacons [Liu et al., 2004, Kandhalu et al., 2010].

The neighbour discovery procedure of Bluetooth uses NDREQs and NDREPs. Bluetooth's neighbour discovery procedure is also *asymmetric* in that Bluetooth devices in transmission range of each other need to be in different but compatible states at the same time in order to detect each other [SIG, 2010]. For illustrative purposes the ten major Bluetooth states are shown in Figure 2.1 with the two important states required by Bluetooth for neighbour discovery highlighted. Bluetooth devices in the inquiry state broadcast NDREQs and listen for NDREPs, whereas Bluetooth devices in the inquiry scan state listen for NDREQs and send NDREPs.

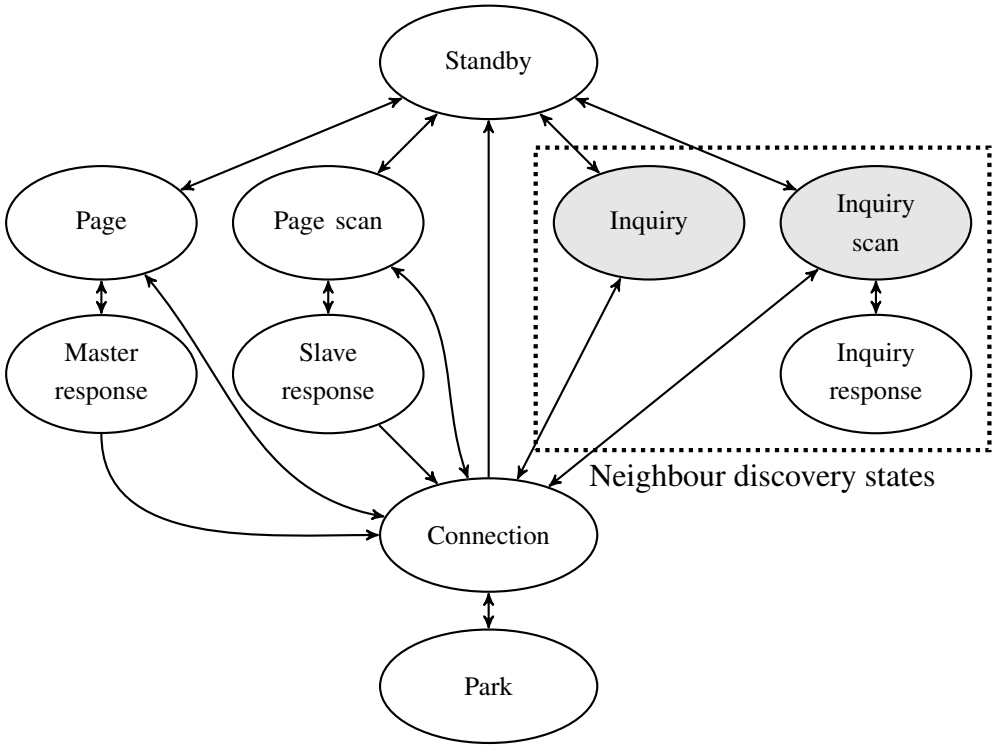



Figure 2.1: Bluetooth 4: Basic Rate state diagram showing the asymmetric neighbour discovery states.

In *symmetric* neighbour discovery there is a single neighbour discovery state in which each device is required to support transmission, reception, and the processing of NDREQs and NDREPs or beacons. For example, the IEEE 802.15.4 standard defines the Medium Access Control (MAC) and Physical Layer (PHY) protocol used in ZigBee [Zare et al., 2013]. It outlines a procedure for symmetric state neighbour discovery between Full-Function Devices (FFDs) using periodically broadcast beacons. Symmetric state neighbour discovery is more commonly used in wireless networks where it is difficult to ensure that devices are in complementary neighbour discovery states when in transmission range of each other [Yang et al., 2009]. Such as when detecting the relationships of Zebra [Liu et al., 2004], or in mountain rescue scenarios [Huang et al., 2005].

As devices in mobile wireless networks are often expected to operate independently for a number of days, weeks, or even years, the key challenges for symmetric neighbour discovery protocols are simplicity, autonomy, low power, speed, and reliability [Bougard et al., 2005, Dutta and Culler, 2008, Bakht and Kravets, 2010].

In order to save energy, *time slot based* symmetric neighbour discovery procedures are used to keep radios powered off for most of the time, yet still guarantee short overlapping active time slots where radios are turned on. One example of a time slot based

| **x** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $v_i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| $v_j$ | - | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

Table 2.1: An example of overlapping active time slots at $x = 6$ and $x = 21$ when two PMWDs $v_i$ and $v_j$ choose relative co-primes 3 and 5 respectively.

approach is called Disco [Dutta and Culler, 2008]. Disco uses the simultaneous congruences shown in Equation 2.1 to guarantee that there will be at least one overlapping active time slot if two PMWDs $v_i$ and $v_j$ choose relative co-primes $p_i$ and $p_j$. For example, Table 2.1 shows that $v_i$ and $v_j$ share common time slots where radios are turned on, despite both devices counting the passage of time from different time slots $a_i$ and $a_j$.

$$\begin{aligned} x &\equiv a_i \ (mod\ p_i) \\ x &\equiv a_j \ (mod\ p_j) \end{aligned} \tag{2.1}$$

The discovery latency (which is defined as the time it takes to discover other devices in the same state that are also in transmission range) in the slot based approach adopted by Disco is variable and depends on the duty cycle. The duty cycle represents the percentage of time that a device is active, e.g. one active slot every 10 slots gives a 10% duty cycle. Generally a lower duty cycle equates to lower power consumption of the device, so in most energy constrained applications a lower duty cycle is desirable. A duty cycle of 10% with Disco will detect 75% of all possible encounters within 2.61 seconds, and 93% within 5.36 seconds. However, when the duty cycle is 5% Disco has a worst case discovery latency of 36 seconds. A newer approach called U-connect offers shorter discovery latency than Disco, and all encounters can be detected within 2.55 seconds with a duty cycle of 1.5% due to shorter slot duration [Kandhalu et al., 2010].

### 2.1.1 Synchronised symmetric neighbour discovery intervals

Instead of the irregular active time slots of Disco and U-connect, another way that PMWDs can save energy yet still discover each other autonomously is to initiate regular symmetric *neighbour discovery intervals* simultaneously on each device in between long periods of sleep [Ye et al., 2002, Liu et al., 2004]. Figure 2.2 illustrates an example where symmetric neighbour discovery intervals are *synchronised* to occur at the same time on two devices.

In Impala [Liu et al., 2004] and CenWits [Huang et al., 2005], GPS-aided time calibration and a regular operation schedule are used to synchronise symmetric neighbour discovery intervals. The benefits of using synchronised symmetric neighbour discovery intervals over a time slot approach include:

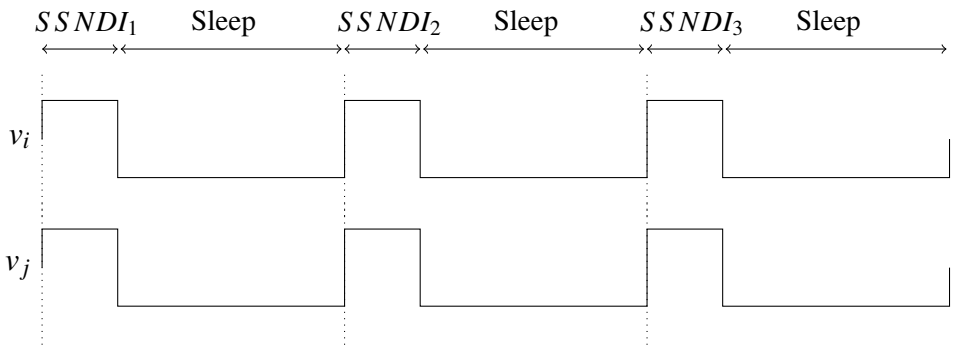



Figure 2.2: The Synchronised Symmetric Neighbour Discovery Intervals (SSNDIs) of two PMWDs.

1. Devices do not need to wait for time slots to overlap, so discovery latency may be shorter.

2. The time between neighbour discovery intervals may be long depending on the application, allowing for very low duty cycles.

The effects of different procedures that are possible within the synchronised symmetric neighbour discovery intervals are not under evaluation in this thesis, but it is assumed that neighbour discovery intervals can be completed quickly. For example, beacons can be broadcast and processed in the synchronised symmetric neighbour discovery intervals used in Impala within 0.035 seconds. Furthermore, it is also assumed that devices will be able to quickly detect collisions and re-transmit beacons within the same neighbour discovery interval.

### Time between symmetric neighbour discovery intervals

As it is being assumed that low discovery latency is possible within synchronised symmetric neighbour discovery intervals, the next logical problem becomes how to choose the length of time between intervals, which is sometimes referred to as the inter-probe time [Orlinski and Filer, 2012b].

During the design of the CenWits system [Huang et al., 2005], it was calculated that two hill walkers with a maximum transmission range of about 70 meters and that are moving past each other on the same path, have 102 seconds to discover the presence of each other. As a result, CenWits devices were configured to use a static inter-probe time of 90 seconds.

After observing that inter-encounter time distributions differ over time in human movement patterns, Wang et al. concurred with Choi et al. by stating that optimal inter-probe times will vary with time [Wang et al., 2007, Choi and Shen, 2011], and that inter-probe times should be shorter during periods with high encounter arrival rates. STAR was produced in order to meet this requirement but misses a relatively large fraction of short encounters due to long inter-probe times [Wang et al., 2009a].

| | BBR | BLE | Wi-Fi | Disco | U-connect | Impala | CenWits |
|---|---|---|---|---|---|---|---|
| **Neighbour discovery states** | Asymmetric | | Asymmetric or symmetric | Symmetric | | | |
| **Operation schedule** | Undefined | | | | | Synchronised | |
| **Inter-probe time (seconds)** | n/a | | | | | 480 | 90 |

Table 2.2: Characteristics of different neighbour discovery procedures found in some wireless communication protocols. Including Bluetooth 4.0: Basic Rate (BBR) and Bluetooth 4.0: Low Energy (BLE). The operation schedule refers to the time when neighbour discovery procedures should be undertaken, and as such are undefined in all but Impala and CenWits which are bespoke protocols designed for specific applications.

STAR requires detailed encounter statistics in order to dynamically alter inter-probe times and save energy. In contrast, DWARF [Izumikawa et al., 2010] requires no prior knowledge of encounters, but needs to continuously scan for wireless signals in order to initiate neighbour discovery intervals. However, as well as adding to the expense of PMWDs, having a second antenna continuously listening for wireless signals can waste energy when there are no new devices in transmission range.

As well as showing how important short encounters are for network connectivity in PSNs, Chapter 3 describes a novel approach to dynamic inter-probe time calculation called PISTONS [Orlinski and Filer, 2012b]. PISTONS requires no prior knowledge of its working environment, nor does it need to continuously scan for wireless signals. Unlike STAR and DWARF, PISTONS only requires an estimate of the maximum movement speed and transmission range of PMWDs as parameters in order to choose efficient inter-probe times.

### 2.1.2 Measuring encounters between PMWDs

Some opportunistic message delivery protocols [Burns et al., 2005, Grasic et al., 2011] state that they require precise identification of the inter-encounter time between nearby PMWDs in order to function correctly. However, it is not always clear what is meant by the inter-encounter time [Passarella and Conti, 2011]. Similarly, how encounters between PMWDs are measured is important in order to understand the results presented in this thesis, and therefore needs careful consideration.

Encounters between PMWDs have to be measured at each PMWD independently because of the time it takes to send and process signals, and because not all signals will be understood [Madan et al., 2010]. The important temporal events which occur between PMWDs $v_i$ and $v_j$ as a result of the different neighbour discovery procedures can be categorised as:

1. **Encounter start time** ($enc\_start^{n}_{v_i v_j}$). The encounter start time ($enc\_start^{n}_{v_i v_j}$) is the time that a PMWD ($v_i$) first becomes aware that another PMWD ($v_j$) has come into transmission range. The encounter start time can also be described as the time when a beacon or NDREP sent by $v_j$ has been received by $v_i$ and results in a new encounter. The separate encounters with other PMWDs can also be indexed by each PMWD using the variable $n$. N.B., this last point implies that two encounters between the same two PMWDs, and which occur at the same time, may have different values for $n$ on the different PMWDs.

2. **Encounter duration** ($enc\_dur^{n}_{v_i v_j}$). The simplest technique used to estimate encounter duration is to count the number of consecutive beacons or NDREPs received from the same PMWD, and to multiply this number by the inter-probe time [Vu et al., 2010]. To account for situations where PMWDs are temporarily unable to receive beacons or NDREPs, some methods for estimating encounter duration also allow for a certain number of missing beacons or NDREPs to still count toward an encounters' duration [Hui et al., 2005, Leguay et al., 2006, Eagle and (Sandy) Pentland, 2006].

3. **Encounter end time** ($enc\_end^{n}_{v_i v_j}$). One technique which can be used to judge an encounter's end time is to continuously monitor the strength of transmitted signals to tell when a PMWD leaves transmission range [Izumikawa et al., 2010]. However, radios in networks made up of battery powered PMWDs should be turned off whenever possible to save energy [Bougard et al., 2005]. Therefore, counting successful consecutive neighbour discovery beacons or NDREPs and estimating the end of encounters when one or more fails to materialise is a more appropriate mechanism with which to calculate encounter duration in PSNs.

The inter-human encounters studied in this thesis are mostly provided by the reality mining datasets described in Section 1.7.1. These empirical datasets contain the start and end times of encounters between participants that were calculated using one of the methods just described. The exception is in Chapter 3, where an artificial encounter starts immediately when one PMWD receives a beacon from a nearby PMWD during neighbour discovery simulations. Moreover, a simulated encounter in the neighbour discovery experiments in Chapter 3 will end as soon as the encountered PMWD leaves transmission range. This is because the ONE simulator creates bi-directional data connections between simulated PMWDs that are broken as soon as PMWDs move out of transmission range of each other.

## 2.2 Ad hoc network formation

Building upon the previous section's description of how PMWDs can discover the presence of others that have come into transmission range, this section will briefly discuss how *ad hoc networks* are formed which enable data to be sent between PMWDs.

Ad hoc networks attempt to provide a reliable, temporary, infrastructure with which participants can establish end-to-end connections for data exchange between themselves. Ad hoc networks are used in a spontaneous manner for only as long as the network is needed, and involve no formal infrastructure. To help with this discussion existing ad hoc network formation techniques are split into two categories.

1. **Manual formation**, where the structure of the network is defined manually by the user, or some user interaction is needed to connect devices.

2. **Autonomous self configuring ad hoc networks**, where devices are assigned addresses and network structures are generated without user interaction.

Wi-Fi devices can form ad hoc networks called Independent Basic Service Sets (IBSSs), with which they can send data to each other without a controlling access point. Wi-Fi devices form IBSSs by passively scanning for any network advertisements, and actively sending beacons containing a wild card network identification parameter. This is referred to in the 2007 Wi-Fi specification as "transmitting probe requests containing the wild card Service Set Identifier (SSID)" [Kerry, 2007].

The Bluetooth specification also allows for ad hoc networks through the formation of piconets. Piconets form using a star topology where Bluetooth devices cooperate to dynamically allocated the role of master to one device, with other nearby devices assigned the role of slaves. Piconets define a *hierarchy* in the network, and data sent between slaves has to go via the master. Additions to Bluetooth also support two or more piconets merging to form larger ad hoc networks called scatternets [Petrioli et al., 2004, Lin and Wang, 2010].

Using commercially available mobile phones, ad hoc network formation using both Wi-Fi and Bluetooth involves some input from the user. Wi-Fi usually comes with software restrictions put in place by manufactures that need to be circumvented before ad hoc connections can be established [Wirtz et al., 2012], and Bluetooth often requires that users "pair" devices using a password before they can communicate. Obviously, any manual configuration each time PMWDs move out of, or into, transmission range of others is simply not feasible for PSNs – network formation should be fully autonomous.

Another issue with current ad hoc network formation techniques is the time needed to determine efficient routes for data to follow through the network before data is sent, and the stability required to maintain the routes. For example, any disruption caused by a master suddenly leaving a piconet results in a new piconet needing to be formed. This process begins with devices forming new connections with each other, which in Bluetooth 4.0: Basic Rate takes 8 seconds on average [Bohman et al., 2004].

The current ad hoc networking capabilities of Bluetooth and Wi-Fi are unsuitable for opportunistic message delivery in highly volatile networks such as PSNs. With that in mind, a new IEEE task group called IEEE 802.15 TG8 has been established with the aim of producing a new specification for low power Peer Aware Communication (PAC) that can be both fully autonomous and support movement speeds of up to 100 km/h. This ongoing effort is currently in the pre-proposal stage, meaning that recommendations are still being made for the future direction of the 802.15.8 PAC standard.

| Protocol | Category | Basis of forwarding decisions | Message replication | Overheads | Delivery probability |
|---|---|---|---|---|---|
| **Epidemic** | Reactive | None (floods the network) | Yes | High | High |
| **Spray and Wait** | Reactive | Number of replicas | Yes | Moderate | Moderate |
| **MobySpace** | Proactive | Co-location | Yes (with MobySpace Spray) | Low | Moderate |
| **MoVe** | Reactive | Velocity | No | Low | Low |
| **Car** | Proactive | Previous encounters and encounter frequency | No | Low | Low |
| **PRoPHETv2** | Proactive | Recent encounters | Yes | Moderate | Moderate |
| **Bubble** | Proactive | Centrality and clusters calculated from previous encounters | Yes | Low | Low |

Table 2.3: Overview of some opportunistic message delivery protocols which can be used in PSNs.

## 2.3 Opportunistic message delivery for PSNs

Route discovery protocols are used for data exchange in computer networks when the path between source and destination is unknown. However, route discovery protocols such as OLSR [Jacquet et al., 2001] and AODV [Perkins and Royer, 1999] are ineffective as a method for data exchange in intermittently connected networks such as PSNs [Spyropoulos et al., 2008]. This is due to routes between devices changing more often than they are discovered, or there being no end-to-end routes at a particular time [Whitbeck and Conan, 2010].

Rather than wasting time, bandwidth, and energy trying to establish stable end-to-end routes which may not exist, *opportunistic message delivery protocols* are being developed that will facilitate multi-hop message delivery in PSNs. Opportunistic message delivery follows the store-and-forward paradigm found in DTNs as messages can be stored until an encounter with a remote PMWD occurs; at which point the remote PMWD may be given a copy of a message, or a message can be transferred to the remote PMWD using a "custody transfer" protocol [Birrane III et al., 2011]. A selection of opportunistic protocols offered in Table 2.3 is presented to illustrate the many different mechanisms with which opportunistic message forwarding decisions in PSNs can be made. Common mechanisms can be categorised as:

1. **Oblivious flooding.** The Epidemic protocol simply copies messages to every encountered device in order to maximise delivery probability [Zhang et al., 2006]. However, flooding the network with copies of messages does not scale well as bandwidth, storage space, and energy is wasted during message duplication. Possible extensions to Epidemic include Spray and Wait which imposes a limit on the number of copies of a message [Spyropoulos et al., 2005].

2. **Probabilistic forwarding.** Many approaches exist with which to calculate the probability of delivering a message via an encountered PMWD. Probabilistic protocols are mostly proactive in that they attempt to gather information about their environment before making forwarding decisions and sharing information with others [Hui et al., 2007a, Musolesi and Mascolo, 2009, Grasic et al., 2011], but some make forwarding decisions based only on the most recent information [LeBrun et al., 2005]:

   (a) **Spatial forwarding.** These methods calculate the probability that a device will be near to a message's destination in the near future. For example, MobySpace uses the frequency of visits to the same Wi-Fi access points when calculating delivery probability [Leguay et al., 2007]. The MoVe protocol uses movement speed and direction to predict where a device is going, and if locations along its route are likely to coincide with the destination's [LeBrun et al., 2005].

   (b) **Encounter based forwarding.** The opportunistic forwarding protocols PRoPHETv2 [Grasic et al., 2011] and Car [Musolesi and Mascolo, 2009] use previous encounters between devices in order to estimate delivery probability. This method is based on the assumption that devices which have met in the past will meet again in the future [Burns et al., 2005].

3. **Data mules.** In Message Ferry [Zhao et al., 2004] and Mobile Ubiquitous LAN Extension (MULE) [Shah et al., 2003] a number of "data mules" provide message relay services for devices which may need to deviate from their normal behaviour in order to rendezvous with mules. Data mules have limited use in PSNs as altering the natural movement patterns of humans for the purpose of message transfer is undesirable.

4. **Cluster based forwarding.** These proactive techniques partition the network using one of many distributed cluster detection algorithms [Daly and Haahr, 2007, Hui and Crowcroft, 2007, Orlinski and Filer, 2012c]. Copies of messages are then given to PMWDs in the same cluster as the destination based on the intuition that people from the same cluster are more likely to meet than those in different clusters [Hui et al., 2007b]. Clustering can act as a means of limiting the epidemic spanning tree needed to reach the message destination where end-to-end paths are difficult or impossible to calculate using distributed algorithms. Another way of describing the effects of cluster based forwarding is that clusters act as boundaries within which to spray copies of a message.

### 2.3.1 Message duplication

Multiple copy opportunistic message delivery protocols such as PRoPHETv2, Epidemic, and Spray and Wait send duplicates of messages in order to increase delivery probability at the expense of efficiency [Spyropoulos et al., 2008]. Protocols that do

not duplicate messages such as Car and MoVe are generally more efficient, and can also prevent messages being dropped by PMWDs due to memory and bandwidth constraints [Lindeberg et al., 2012].

Before sending copies of messages, PRoPHETv2 uses the history of previous encounters to estimate the probability that an encountered device can deliver a message. PRoPHETv2 is a well established protocol from which many key insights about opportunistic message delivery are being discovered. For example, it was shown in the Networking for Communications Challenged Communities (N4C) experiment that some of the parameters PRoPHETv2 needs to function correctly (typical inter-encounter times and a suitable constant for delivery probability ageing [Grasic et al., 2011]) are difficult to calculate for different areas of the network. N4C also highlighted the "parking lot problem" where many short encounters are separated by short time periods whilst human movement patterns dictate longer durations. The parking lot problem was caused by poor Wi-Fi connections [Grasic et al., 2011], and is one of the reasons why cumulative and not single encounter duration is being used in the new clustering algorithms described in chapters 4, 5, 6 and 7.

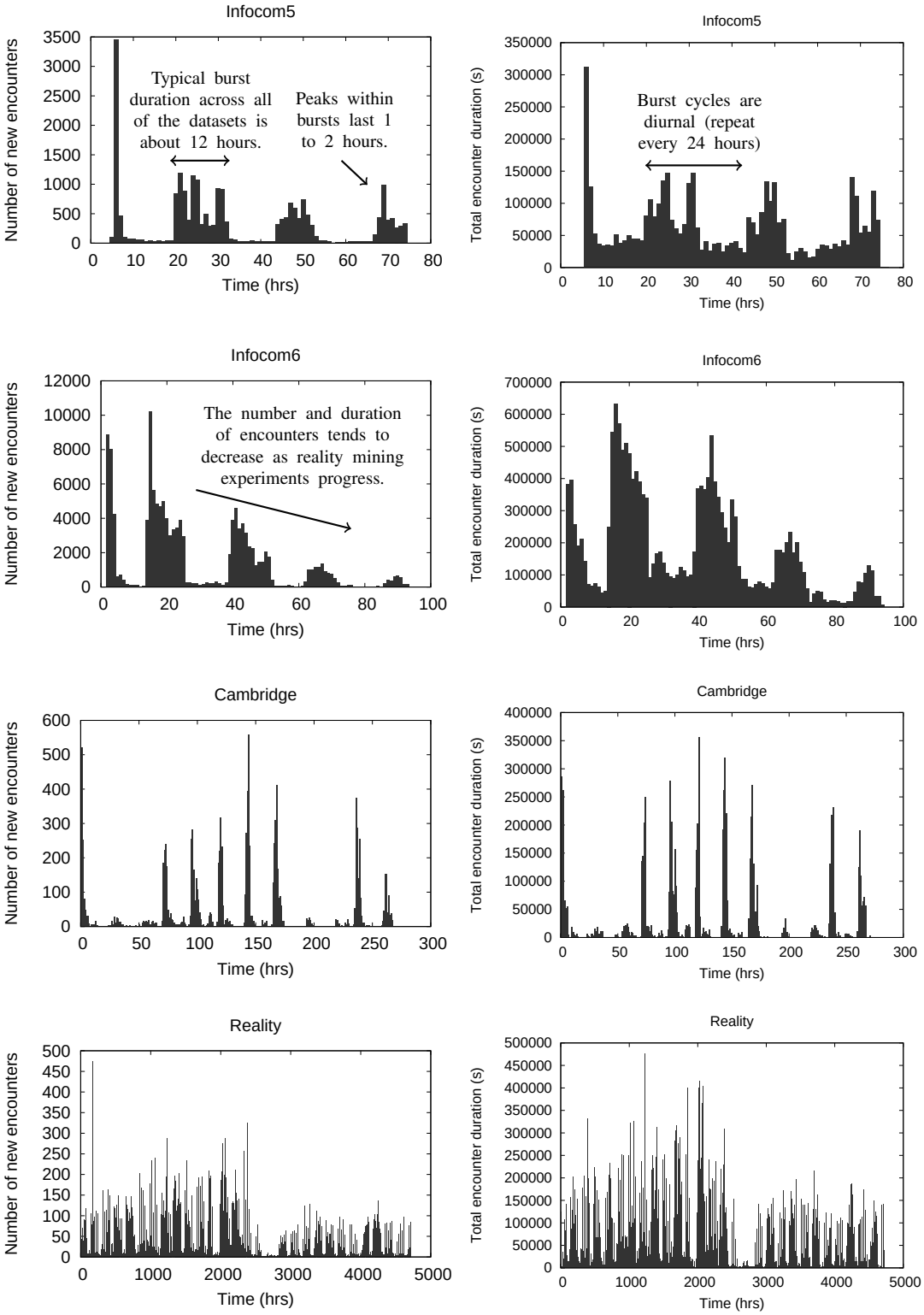
## 2.4 Human encounter patterns

Data from discrete chronological time frames can be plotted in order to visualise the collective encounter patterns present in reality mining datasets. For example, the time series of new encounters and encounter duration depicted in Figure 2.3 show bursts of activity lasting for around twelve hours [Wang et al., 2009b], with peaks of activity lasting no longer than two hours common within bursts.

Encounter patterns differ between participants of reality mining experiments, but individual encounter patterns repeat on a daily basis [Henderson et al., 2008]. The diurnal encounter patterns also dictate that the probability of a meeting between participants differs between time frames shorter than 24 hours [Chaintreau et al., 2005, Ekman et al., 2008], and changes to measurements such as the shortest and fastest paths between participants have been seen when analysing encounters from rush hour and non-rush hour traffic [Leung et al., 2011].

By using reality mining data and vector clocks it is possible to calculate the minimum time needed to send a message from one participant to another using opportunistic message delivery protocols (also known as the temporal distance) [Fidge, 1988, Pan and Saramäki, 2011]. A generalisation about how quickly one PMWD can be reached from another can also be calculated using Pan and Saramäki's equation shown in Equation 2.2, where $\zeta_{v_i v_j}$ is the average temporal distance between $v_i$ and $v_j$ and $n$ is the total number of PMWDs [Pan and Saramäki, 2011].

$$F_t(v_i) = \frac{1}{n-1} \sum_{j=1}^{n} \frac{1}{\zeta_{v_i v_j}} \tag{2.2}$$




(a) New encounters per hour (b) Hourly total encounter durations

Figure 2.3: New encounters and total encounter durations in hourly time frames for the Infocom5, Infocom6, Cambridge and Reality datasets.

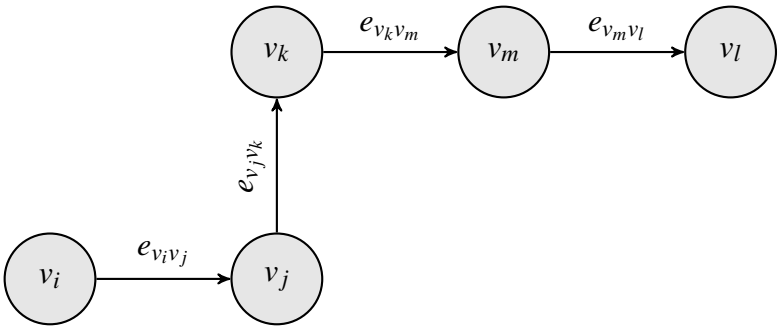



Figure 2.4: A static graph showing relationships (edges) between five vertices.

Recent work has shown that temporal measurements in graphs created from reality mining data (such as vertex lifespan, edge attachment rate by vertex age, and inter-edge temporal gaps) differ between vertices [Allamanis et al., 2012]. The next section will discuss in more detail how reality mining datasets can be analysed in graph form.

## 2.5 Encounter graphs

This section discusses how encounters between PMWDs in reality mining data can be analysed using graphs. A distinction is made between monotonic encounter graphs where the encounter information is presented as a single evolving graph, and dynamic encounter graphs [Zhou et al., 2007, Scellato et al., 2011a, Pan and Saramäki, 2011] where the sequence of temporal events in the data can be preserved.

### 2.5.1 Static graphs

Social and communication networks are commonly depicted using static graphs such as the one pictured in Figure 2.4. Generally speaking, an edge in a static graphs is used to represent the result of one or more interactions between two vertices, or some potential of the interactions between the vertices. Two examples of quantifiable measurements between PMWDs that edges can express are communication bandwidth and cumulative encounter duration.

In cases where static graphs are being used to represent cumulative data, the weight of edges may have been calculated from many separate interactions. However, important information such as the distribution of individual interaction weights or the timing of interactions may be lost when there are many separate interactions that contribute to the same edge.

A partial strategy for preserving the temporal information when modelling information using graphs is described in Section 2.5.3. This approach simply separates data into discrete chronological time frames (or “chunks” [Bilmes, 2010]), allowing the analysis of a separate static graph in each time frame.

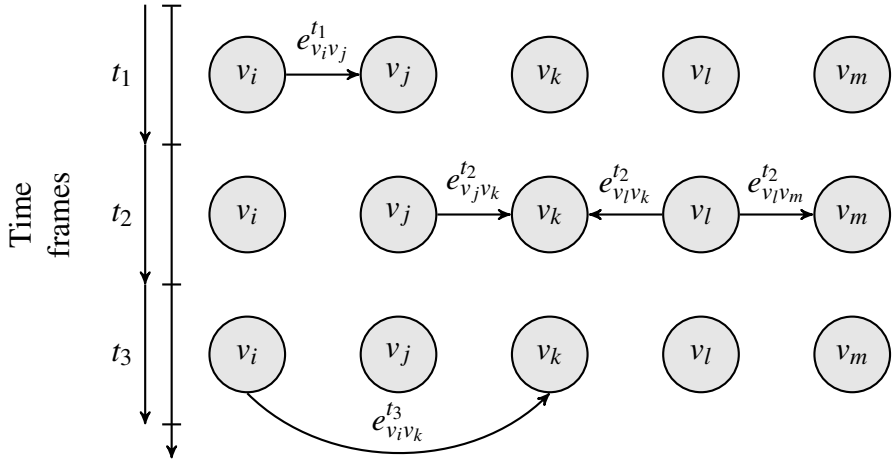



Figure 2.5: A dynamic encounter graph between time frames $t_1$ and $t_3$. Each time frame contains an independent static graph with directional edges between vertices.

### 2.5.2 Monotonic encounter graphs

In communication networks and many other applications of graph analysis, graphs are subject to change as vertices or edges are added and removed over time.

In order to distinguish between the static graphs described in Section 2.5.1 and graphs which can grow over time, a graph to which vertices can only be added and edges can only be added or updated is referred to as a monotonic graph. This thesis will not discuss the formal concept of fully dynamic graphs where unrestricted insertions and deletions of edges and vertices are permitted [Demetrescu et al., 2005].

*Monotonic encounter graphs* are used in the analysis of reality mining datasets, and are created using a process whereby encounter information is chronologically added to a graph one encounter at at time. The distinction between static and monotonic graphs is especially important when discussing the rate at which monotonic encounter graphs grow as edges are added to them in Section 5.3.

### 2.5.3 Dynamic encounter graphs

In order to analyse the network topology created by transient encounters between PMWDs, dynamic encounter graphs such as the one pictured in Figure 2.5 can be used instead of a single static or monotonic encounter graph [Williams et al., 2012].

Within dynamic encounter graphs there are a number of vertices which do not change between discrete time frames, and pair-wise observations (edges) within time frames are independent of observations from other time frames. This results in a series of chronologically indexed static graphs, one static graph for each time frame. For example, an edge representing a directional encounter between vertices $v_i$ and $v_j$ within time frame $t_1$ is illustrated in Figure 2.5 as an arrow labelled $e^{t_1}_{v_i v_j}$. If the encounter between $v_i$ and $v_j$ spanned both $t_1$ and $t_2$ then it would have been illustrated as two

separate edges (one edge between $v_i$ and $v_j$ in $t_1$, and another in $t_2$).

Suppose that edges in the dynamic encounter graph shown in Figure 2.5 represent directional data connections between PMWDs. Then one phenomenon of dynamic encounter graphs that is not present in static or monotonic encounter graphs is that the shortest path between PMWDs may not be quickest [Panisson et al., 2011]. Moreover, the period of observation in the available reality mining datasets is finite. Thus the total number of future encounters and remaining time frames in monotonic and dynamic encounter graphs decreases as time increases, and the probability of a path existing between two PMWDs also decreases [Pan and Saramäki, 2011].

Identifying a meaningful resolution for dynamic encounter graphs is critical to matching the rate of change in the network topology [Sulo et al., 2010]. Even when using a very high resolution to create the discrete time frames (resulting in many short time frames), temporal information about interactions between vertices can still be lost. This is because within each time frame there is a static graph similar to those described in Section 2.5.1, and within which single edges may represent a simplification of many separate interactions between vertices.

However, longer time frames offer a convenient way of analysing PSNs which helps to mitigate the parking lot problem and produce larger clusters of PMWDs (which is shown in Chapter 4 to increase opportunistic message delivery probability). Mitigating the parking lot problem is done by using edges that represent the cumulative encounter duration between PMWDs within each time frame. Thus giving a metric which can be used to determine for how long pairs of PMWDs were connected to each other despite the frequent disconnections seen as a result of the parking lot problem.

## 2.6 Graph analysis

A number of different techniques exist for analysing static, monotonic, and dynamic encounter graphs. This section will detail analysis techniques which are used in this thesis to describe reality mining data in static graphs, as well as describing techniques which can reveal how reality mining data changes over time in monotonic graphs.

### 2.6.1 Static analysis

The following analysis techniques can be used to describe static graphs or to analyse monotonic graphs at any point during data aggregation. In each of the analysis techniques discussed PMWDs are represented as vertices, and edges are used to represent either directional or bi-directional relationships.

#### Density

Graph density can be used to quantify the connectivity of a static graph by calculating the number of edges in proportion to the number of vertices. A dense graph would be one in which the number of edges ($|E|$) is close to the maximum dictated by the number

| | Infocom5 | Infocom6 | Cambridge | UCL1 | Reality |
|---|---|---|---|---|---|
| **Environment** | Conference | | Campus | | |
| **Duration (days)** | 3 | 3 | 12 | 6 | 246 |
| **Number of vertices** | 41 | 78 | 36 | 20 | 97 |
| **Number of edges after aggregation** | 1568 | 5813 | 1033 | 135 | 5167 |
| **Density** | 0.9561 | 0.9679 | 0.8198 | 0.3553 | 0.5549 |

Table 2.4: Aggregate density of some reality mining datasets.

of vertices ($|V|$). The opposite of a dense graph is a sparse graph in which there are very few edges in relation to number of vertices.

A directed graph such as that which is generated from encounters between PMWDs has $|V|(|V|-1)$ maximum edges and a density calculated using Equation 2.3.

$$\frac{|E|}{|V|(|V|-1)} \tag{2.3}$$

Density can be used to measure how many PMWDs encountered each other during reality mining experiments. Predictably, the density values given by Equation 2.3 and shown in Table 2.4 show that the campus wide experiments have a lower density than experiments conducted as conferences. This difference in density is intuitive; campus wide experiments are usually conducted over larger geographic areas, therefore PMWDs will be more widely distributed than at a conference.

### Centrality

Centrality is used in Chapter 3 to test the effects of different inter-probe times on the connectivity of PMWDs. Degree centrality is the simplest form of centrality, it is defined as the number of edges incident upon a vertex. Or when applied to PSNs, degree centrality of a PMWD can refer to the number of bi-directional data connections made to nearby PMWDs.

Betweenness centrality is a measure which represents the fraction of shortest paths from all vertices to all others that pass through that vertex. Therefore, betweenness centrality can be used to identify the most connected PMWDs in the same way that Freeman used it to quantify the level of control humans have on peer-to-peer communication in social networks [Freeman, 1977].

In the betweenness centrality calculation shown in Equation 2.4, $\sigma_{v_j v_k}$ is the total number of shortest paths from random vertex $v_j$ to random vertex $v_k$ and $\sigma_{v_j v_k}(v_i)$ is the number of those paths that pass through $v_i$. The sum over all random pairs of vertices is taken and normalised to give the final betweenness centrality ($Cen$) of vertex $v_i$.

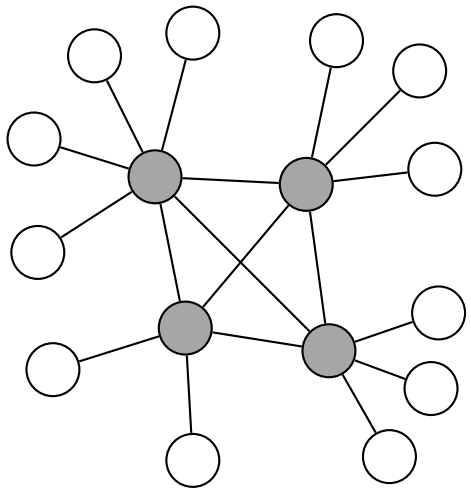


Figure 2.6: A small world network with vertices with higher betweenness centrality highlighted.

$$Cen(v_i) = \sum_{v_j \neq v_i \neq v_k \in V} \frac{\sigma_{v_j v_k}(v_i)}{\sigma_{v_j v_k}} \tag{2.4}$$

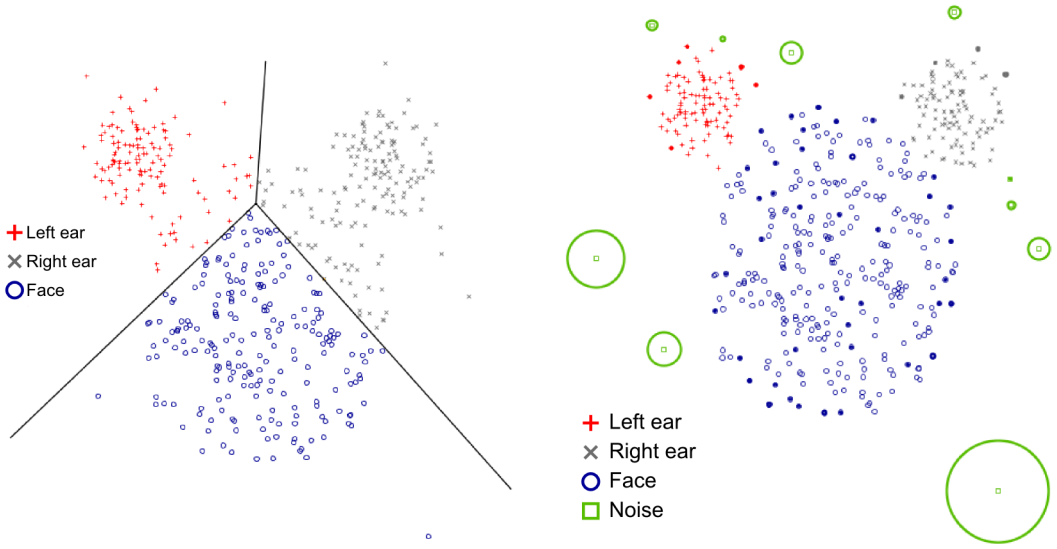
In directed encounter graphs, out-degree centrality measures the number of directed edges that a vertex directs to others and is a measure of the gregariousness of vertices. In contrast, in-degree centrality is a separate measure which gauges the popularity of a vertex. PMWDs in reality mining datasets are heterogeneous in terms of gregariousness and popularity. In fact, reality mining datasets are examples of small world topologies [Easley and Kleinberg, 2010], such as is illustrated in Figure 2.6. Within small world topologies there is a large difference in degree centrality and betweenness centrality among vertices, but paths between any two vertices are small relative to the size of the network [Daly and Haahr, 2007].

### 2.6.2 Densification exponent

Density and centrality can be used to comment on the resulting data collected by reality mining experiments, but in order to see how data changes over time a different approach is needed.

Collectively, the vertex and edge attachment rate of individual vertices can cause a monotonic encounter graph to grow and become more connected over time. When encounter graphs grow, they may grow to a specific Densification Power Law (DPL) exponent [Leskovec et al., 2005].

$$e(t) \propto v(t)^{\kappa} \tag{2.5}$$




(a) Partition clustering.

(b) Spatial clustering.

Figure 2.7: Analysis of the mouse dataset (so called because it looks like a mouse) using two different clustering algorithms available in ELKI [Achtert et al., 2012].

The DPL exponent shown in Equation 2.5 states that the number of edges over time ($t$) raised to some power $\kappa$ is proportional to the number of vertices. The higher the DPL exponent $\kappa$ is, the faster additional edges are added to the encounter graph compared to new vertices. The edge attachment rate of reality mining data is shown to closely follow a DPL later in Section 5.3.

## 2.7 Clustering

Cluster detection has been an essential part of a data analyst's toolkit ever since Sokal and Sneath first refined it in the 1960s [Sokal, 1963]. Since then many more clustering techniques have been defined, and modern cluster detection algorithms have revealed new insights such as for how long people gather in transient groups when on university campuses or during conferences [Orlinski and Filer, 2013].

The basic idea of clustering is to allocate each of $n$ objects to $k$ different clusters; a cluster can often be identified as a group of vertices that have more edges among each other than with other vertices of a graph [Newman, 2004]. There are many cluster detection methods which can be applied to reality mining data analysis, these include but are not limited to:

1. **Partitional clustering.** Decomposes vertices directly into disjointed clusters that optimise some chosen partition criteria. The number of clusters is usually specified [MacQueen, 1967]. For example, the k-means clustering algorithm with $k = 3$ will split the data points in Figure 2.7a into 3 disjointed clusters [Lloyd, 1982].

$$\begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{cccc} 1 & \dots & j & \dots & n \end{array} \\ \begin{pmatrix} & & & & \\ & & \vdots & & \\ & \dots & u_{ij} & \dots & \\ & & \vdots & & \\ & & & & \end{pmatrix} \end{array}$$

Figure 2.8: A $k$ by $n$ cluster matrix.

2. **Density based clustering.** Creates arbitrary shaped clusters based on vertex density [Agrawal et al., 1998, Ester et al., 1996], such as the clusters detected by OPTICS [Ankerst et al., 1999]. Unlike partitional clustering methods, density based clustering can produce outliers (data points that do not belong to any clusters).

3. **Spatial based clustering.** Commonly used in spatial data mining and in grid data structures [Wang et al., 1997, Schikuta and Erhart, 1997]. Euclidean distance is being used in the clustering algorithm from Lu et al. [Lu et al., 2003] in Figure 2.7b. Figure 2.7b also shows the outliers that are geographically distant from all cluster centres.

4. **Budget-based clustering.** Generates clusters using one of many agglomerative clustering algorithms, with the addition of an upper bound which specifies maximum cluster size [Krishnan and Starobinski, 2006].

The clustering shown in Figure 2.7 is strict i.e. no vertex belongs to more than one cluster. However, cluster partitioning can also be *fuzzy*, i.e. vertices can belong to one or many clusters. Fuzzy clusters can be represented as a matrix such as the one shown in Figure 2.8. The value $u_{ij}$ within the matrix is the membership function which is used to express the degree of belonging of vertex $j$ to cluster $i$ [Trauwaert, 1988]. The value one is used to express complete belonging, or zero to express absence of any kind of belonging. Equations 2.6 and 2.7 relating to this matrix must always be satisfied, but when all $u_{ij}$ are either zero or one, the cluster is strict, otherwise it is fuzzy.

$$0 \leq u_{ij} \leq 1 \quad \forall i, j \tag{2.6}$$

$$\sum_{i=1}^{k} u_{ij} = 1 \quad \forall j. \tag{2.7}$$

As well as clusters that are detected centrally, PMWDs can detect their own *local clusters* using distributed algorithms. Local clusters are an individual's view of the cluster to which they belong held in local memory. The next subsection will talk about

some existing distributed clustering algorithms, most of which will produce fuzzy local clusters.

### 2.7.1 Distributed cluster detection algorithms for PSNs

All of the clustering algorithms introduced so far in Section 2.7 are centralised i.e. they require access to all of the data in the dataset in order to function correctly. In distributed cluster detection algorithms for PSNs, PMWDs are responsible for clustering themselves without access to a centralised data store [Hui and Crowcroft, 2007, Orlinski and Filer, 2012c].

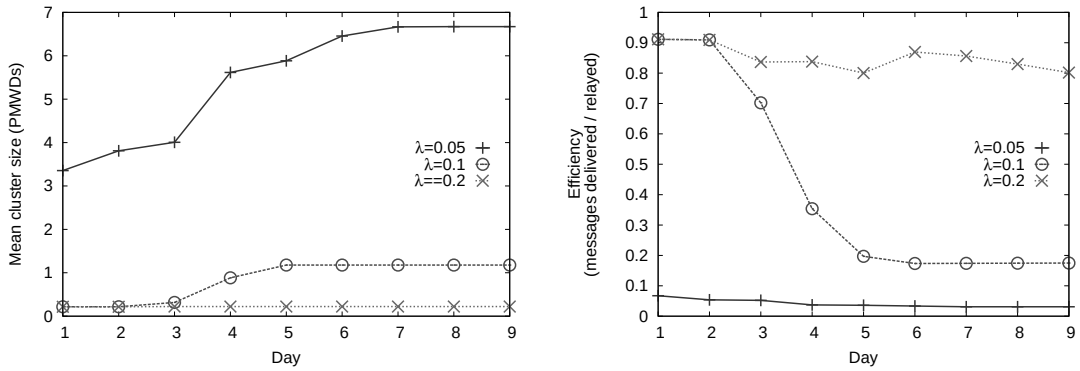
In some cases, the social relationships between PSN participants can be obtained in advance. Clusters of social PMWDs can then be extracted using a centralised clustering algorithm, and each PMWD allocated a cluster label [Hui and Crowcroft, 2007]. This approach can also be adopted to bootstrap distributed cluster detection algorithms [Bigwood et al., 2008]. However, in cases where there is no prior relationship information available, or where social relationships do not correspond to encountered PMWDs [Gaito et al., 2012], PMWDs should be able to discover clusters themselves.

One method with which PMWDs can create clusters themselves is to propagate cluster labels through the PSN during encounters [Leung et al., 2009]. However, during epidemic label propagation, a single cluster label can spread across the entire network resulting in the *monster cluster problem*. SHARC [Herbiet and Bouvry, 2010] prevents monster clusters from forming but still suffers from the *wandering cluster problem*, which is caused when large groups of PMWDs propagate their cluster labels elsewhere.

Modularity, K-Clique, and Simple [Hui et al., 2007b] are examples of distributed clustering algorithms in which PMWDs exchange local clusters rather than just cluster labels. This allows PMWDs to receive information about others that they have not encountered directly, and to create clusters that are multiple hops in size. However, Modularity, K-Clique, and Simple all suffer from monotonically increasing cluster sizes [Orlinski and Filer, 2013], which can lead to abnormally large clusters if the algorithms are left to run for long enough. This is because these algorithms aggregate data from all prior encounters in order to cluster PMWDs, without looking at the situational relevance or time passed since the encounters took place.

For example, Simple works by promoting encountered PMWDs to *familiar sets* once their cumulative encounter duration exceeds a *familiar threshold*. The familiar threshold acts as a mechanism with which to separate the *social* PMWDs (those that encounter each other regularly and for long periods) from the *non-social ones* (everyone else). Non-social devices have also been called *vagabonds* elsewhere in the literature [Zyba et al., 2011].

As well as a familiar set, Simple also maintains local clusters on each PWMD which contain all of the PMWDs in the familiar set as well as PMWDs included via the checks outlined in [Hui et al., 2007b]. An important parameter used by Simple to add PMWDs to local clusters is called $\lambda$. Low values of $\lambda$ specified in Equation 2.8 will allow PMWDs with fewer social PMWDs in common to be added into local clusters,




(a) Cluster size over time using Simple. (b) Message delivery efficiency using Bubble.

Figure 2.9: Cluster size and message delivery efficiency (messages delivered / relayed) over time using Simple, and Bubble to deliver messages within the Cambridge reality mining dataset [Scott et al., 2009].

thus lower values of $\lambda$ generally increase local cluster size more quickly than higher values.

**Adding PMWDs to local clusters in Simple.** Add $v_j$ to the local cluster of $v_i$ ($C_i$) if the intersection of the familiar set of $v_j$ ($FS_j$) and $C_i$ divided by the size of $FS_j$ is greater than $\lambda$:

$$|FS_j \cap C_i|/|FS_j| > \lambda \tag{2.8}$$

Local clusters can also be merged with the local clusters of other PMWDs. The decision whether to merge local clusters in Simple is made based on the parameter value $\gamma$ and the intersection of the two local clusters as shown in Equation 2.9. Lower values of $\gamma$ mean that the intersection between local clusters can be smaller before merging. Thus lower values of $\gamma$ encourage local cluster mergers more often, which increases cluster size over time.

**Merging local clusters in Simple.** If $v_j$ is added to $C_i$ after Equation 2.8, then Simple will merge the local clusters of $v_i$ and $v_j$ if:

$$|C_i \cap C_j| > \gamma|C_i \cup C_j| \tag{2.9}$$

Figure 2.9a shows the result of an example where mean cluster size increasing monotonically or not at all with Simple depending on the $\lambda$ value chosen. Using the same local clusters produced in Figure 2.9a, the cluster based message delivery algorithm Bubble [Hui et al., 2007b] exhibits poor efficiency when $\lambda$ is 0.05, changing efficiency when $\lambda$ is 0.1, and good efficiency when $\lambda$ is 0.2. Understanding this relationship between the cluster partitioning, the cluster based message delivery algorithms, and message delivery efficiency is the focus of Chapter 4, which will look more closely at existing *aggregated monotonic clustering* algorithms using the analysis techniques introduced in the following subsection.

### 2.7.2 Cluster analysis techniques

The mouse dataset shown in Figure 2.7 contains three Gaussian clusters and noise, and is arranged to look like a mouse. It is commonly used to illustrate the behaviour of different clustering algorithms [Achtert et al., 2012].

In the same way as different partitions can be produced by the different clustering algorithms using the mouse dataset, many different clusters are possible in reality mining data when using distributed clustering algorithms. The following techniques are used in Chapter 4 to describe the clusters produced using distributed clustering algorithms.

The Jaccard index, also known as the Jaccard similarity coefficient [Jaccard, 1901], assess the similarity of clusters. It can be defined simply as the intersection of two clusters, $A$ and $B$, divided by the size of the union as shown in Equation 2.10. The Jaccard index can be used to assess if clusters share a large number of vertices, and thus if the clustering has separated the data into fuzzy or strict clusters.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.10}$$

To measure the level of mutual inclusion (fuzziness) of $n$ PMWDs between $k$ local clusters, Dunn's partition coefficient [Bezdek, 1981, Trauwaert, 1988] shown in Equation 2.11 can be used. The closer $F_k(C)$ is to one, the more strict or non-fuzzy the cluster partitioning of $C$ where $\frac{1}{k} \leq F_k(C) \leq 1$ and $\mu_{ij}(i = 1,2,...,k; j = 1,2,...,n)$ is the membership function of vertex $j$ in cluster $i$.

$$F_k(C) = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n} (\mu_{ij})^2 \tag{2.11}$$

## 2.8 Summary

This chapter begins with a discussion of how peer aware wireless networks called PSNs can be formed using existing wireless networking technology. Also included in this chapter is a review of some of the important complex network, clustering, and analysis techniques that can be used to describe the human encounter patterns present in reality mining datasets.

Throughout the rest of this thesis the analysis techniques just described are used for a variety of different applications. Such as measuring the level of connectivity in PSNs in Chapter 3, and describing the clusters produced by distributed clustering algorithms in Chapter 4.

Before introducing new distributed cluster detection and opportunistic message delivery algorithms for PSNs in chapters 4, 5, 6 and 7, the next chapter will discuss an orthogonal but equally important issue for data delivery in PSNs. Specifically, Chapter 3 will propose a method with which PMWDs in a PSN can automatically detect encounters.

# Chapter 3

# Neighbour discovery for PSNs

Neighbour discovery is critically important for opportunistic message delivery in PSNs. As well as transmission range and bit rate, message latency in PSNs is constrained by slower movement speeds of participants [Choi and Shen, 2011] and the rate at which PMWDs search for others [Nguyen et al., 2011].

However, energy is a scarce resource for PMWDs, and neighbour discovery is an energy intensive procedure [Perrucci et al., 2011]. Battery powered "throwboxes" used in early DTN experiments used up to 99% of their total energy searching for new contacts [Banerjee et al., 2010]. Therefore, the neighbour discovery procedures used in PSNs need to be as efficient as possible in order to preserve battery life on participants PMWDs.

The aim of this chapter is to theoretically assess the suitability of symmetric neighbour discovery intervals (see Section 2.1.1 for more details) for neighbour discovery in PSNs. This chapter will also propose a means of calculating the time between symmetric neighbour discovery intervals (called the inter-probe time) in such a way that can ensure that encounters between PMWDs are detected reliably, whilst at the same time giving the user the flexibility to specify their desired level of connectivity and/or energy saving.

## 3.1 Detecting encounters

One way of detecting when humans interact with each other is to record the face-to-face conversations between them [Choudhury et al., 2003]. Face-to-face conversations are important for epidemiology [Apolloni et al., 2009] and knowledge propagation in social networks [Madan et al., 2010], but they cannot tell us about the people who are in close proximity to one another but do not interact, e.g. the familiar strangers [Milgram, 1977].

The proximity of PMWDs to one another can be analysed with reality mining datasets, but creating these datasets depends on PMWDs being able to detect when others are nearby, or having some way of recording the geographic location of PMWDs (such as GPS) so that proximity can be inferred using a centralised process.

This chapter will address the problem of proximity sensing in PSNs using neighbour discovery, which is the process whereby PMWDs act independently to discover others in transmission range. The following subsection will briefly explain why further research in this area is needed by discussing the neighbour discovery procedures currently used in Bluetooth and Wi-Fi.

### 3.1.1 Using Bluetooth or Wi-Fi for neighbour discovery in PSNs

Bluetooth was designed primarily as a wire-replacement for low power mobile devices, and can be found in many mobile phones sold today. As a result there are few extra hardware costs associated with using Bluetooth in PSNs. Wi-Fi can provide a longer transmission range and higher bandwidth than Bluetooth, but continuously scanning for Wi-Fi networks can reduce battery life on current devices from longer than 310 hours to only 5 hours [Han et al., 2012a]. In contrast, continuously scanning for nearby devices using Bluetooth reduces battery life on the same mobile phones to around 20 hours; which means users may not need to charge their phones during the daytime if Bluetooth was used as the means of device-to-device communication in PSNs.

However, despite Bluetooth's benefits it is not a complete solution for the autonomous neighbour discovery needed in PSNs. One reason for this is that the neighbour discovery states are asymmetrical as explained in Section 2.1. Asymmetrical states are problematic because the assumption that two PMWDs in transmission range are configured as one having been the sender and the other the receiver at a particular time is not realistic [Yang et al., 2009].

Moreover, Bluetooth devices also need to be in transmission range for long periods of time in order to discover each other. Bluetooth 4.0: Basic Rate devices in the Inquiry state can sometimes wait up to 10.24 seconds (or more in error-prone environments) for a NDREP because of the time needed to perform the 1024 frequency hops outlined in the Bluetooth specification [SIG, 2010]. This means that discovery between stationary devices is often much more reliable than neighbour discovery between mobile devices. In the 2008 MobiClique experiments [Pietilainen et al., 2008], 50% of PMWDs were successfully detected during the day using Bluetooth. Whereas at night success rates go up to nearly 100% because PMWDs were mostly stationary in hotel rooms.

A further issue with the use of Bluetooth for PSNs is that unless a data transfer or discovery request is made by some application acting on behalf of the user, Bluetooth will remain in a low power standby state in order to conserve energy. This behaviour is not suitable for PSNs which would require autonomy in order to create networks made up of many PMWDs always willing to exchange data between themselves.

A more comprehensive solution for neighbour discovery in PSNs is to use the symmetric neighbour discovery intervals that were used in Impala, but significant additions would need to be made to the Bluetooth or Wi-Fi protocols in order to support this functionality. Despite this concern, it is not unreasonable to suggest that the required changes can be made. The IEEE 802.11 standards which outline the functionality of Wi-Fi are currently undergoing a similar process to accommodate data exchange between high-speed vehicles in IEEE 802.11p for VANETs [van Eenennaam et al., 2012].

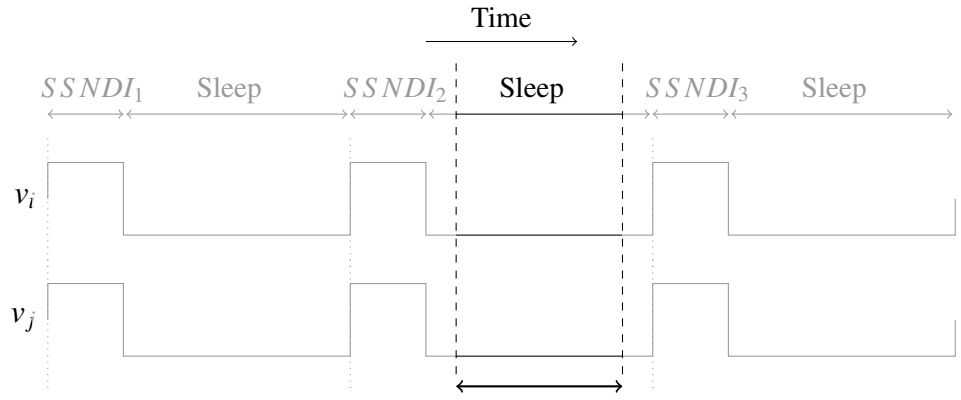



Figure 3.1: An inter-probe time which is too long may lead to some short encounters being missed. In this example the PMWDs $v_i$ and $v_j$ briefly come into transmission range of each other during the period $p_{int}$. There is no Synchronised Symmetric Neighbour Discovery Interval (SSNDI) during $p_{int}$, thus $v_i$ and $v_j$ do not discover each other.

A new IEEE task group has also been established with the aim of producing the IEEE 802.15.8 standard for low power Peer Aware Communications (PAC).

### 3.1.2 Problem statement

The length of inter-probe times is critically important if using synchronised symmetric neighbour discovery intervals as nearby devices cannot be discovered between intervals. If PMWDs are in transmission range with each other for less than the inter-probe time then encounters may be missed entirely as shown in Figure 3.1, or encounters may be part detected as Figure 3.2 shows. This chapter presents a mechanism with which to calculate suitable inter-probe times for PSNs, and discusses the consequences of choosing certain long or short inter-probe times.

A number of strategies can be adopted in order to choose suitable inter-probe times for PSNs. For efficiency, the inter-probe time giving the smallest number of neighbour discovery intervals per new discovery can be used. It has been shown previously that there can be many efficient inter-probe times when measured in this way [Izumikawa et al., 2010]. However, it is not clear how inter-probe time choices will affect other measures of connectivity in PSNs such as centrality and the lengths of encounters. Section 3.4 will look at those measurement in more detail.

It is important to point out that the synchronised symmetric neighbour discovery model presented visually in Figure 3.1 and Figure 3.2 does not take into account lost or partly understood neighbour discovery beacons sent by PMWDs. The exact mechanism by which nearby PMWDs are discovered during neighbour discovery intervals is not discussed further in this thesis. It is assumed that beacons are sent and listened

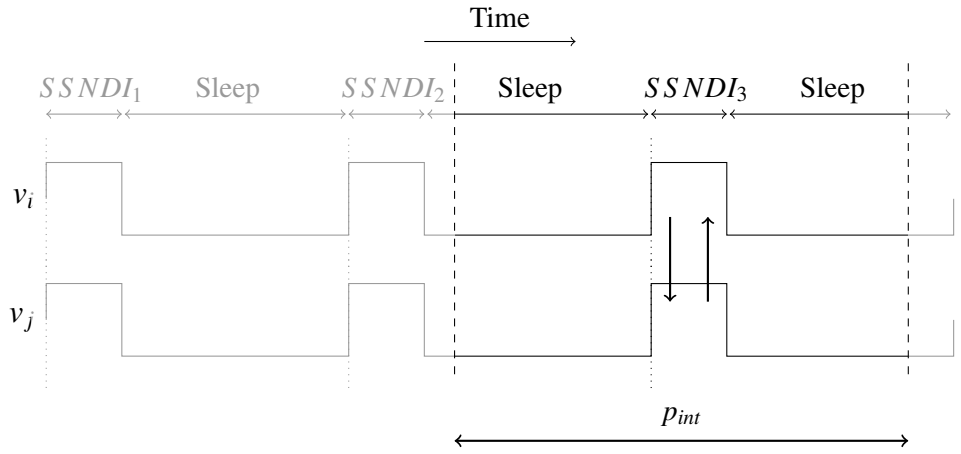



Figure 3.2: In this example the PMWDs $v_i$ and $v_j$ are in transmission range of each other during a different period from Figure 3.1. This time the two PMWDs can exchange beacons within the third pictured SSNDI.

for during symmetric neighbour discovery intervals, and that the length of the interval is sufficient for nearby PMWDs to receive and process the beacons. Lost or misunderstood neighbour discovery beacons are not considered here.

## 3.2 Experimental environment

Reality mining datasets cannot easily be used to validate new neighbour discovery algorithms for reasons that will now be explained. In all of the reality mining experiments shown in Table 1.1 where Bluetooth was used for neighbour discovery, the time between devices entering the Inquiry state was at least 120 seconds. It is not possible to lower this retrospectively in order to infer missing encounters without using some other data such as GPS. However, GPS data is often missing for long periods where PMWDs were indoors, and reliance on location data may lead to inaccurate assumptions about connectivity between PMWDs in noisy or error-prone environments.

Synthetic movement models allow experimentation with higher a degree of fidelity and participation than is found in existing reality mining datasets. The synthetic movement models that are used in neighbour discovery experiments in this chapter are the Working Day Movement (WDM) [Ekman et al., 2008] and Random Walk Movement (RWM) [Keranen et al., 2010] models.

The inter-encounter times of the RWM model are distinct from those of the WDM model in that they are light-tailed. PMWDs moving to the RWM model also move independently and are identically distributed in bounded regions [Sharma et al., 2007]. WDM is used for more realistic pedestrian movement as PMWDs following this model spend long periods being stationary in “home locations” [Henderson et al., 2008] and are not evenly distributed geographically.

| | University | Manhattan | Helsinki |
|---|---|---|---|
| **Number of simulated PMWDs** | 235 | 200 | 200 |
| **Area size (m)** | 380 x 265 | 6000 x 5600 | 7200 x 6800 |
| **Transmission range (m)** | 10 | | |
| **Days** | 2 | | |

Table 3.1: Simulation maps and number of participants used in the different neighbour discovery experiments.

For the experiments in this chapter both the RWM and WDM movement models will be used in conjunction with the three maps described in Table 3.1. These maps limit simulated PMWDs to paths found in real life environments, and the ONE simulator [Keranen et al., 2009] provides a means of altering the inter-probe times of the simulated PMWDs.

The Manhattan and Helsinki maps are bundled with the ONE simulator. Both have been used extensively in the Connectivity, Applications, and Trials of Delay-tolerant Networking (CATDTN), and the Security Infrastructure for DTN (SINDTN) projects, and both have been confirmed to work correctly with the RWM and WDM models [Ekman et al., 2008]. The streets in the Manhattan map are laid out in rectangular blocks, whilst the winding streets in the Helsinki map are identical to those found in downtown Helsinki. Both the Manhattan and Helsinki maps are used in this chapter to test neighbour discovery protocols for PSNs in different realistic urban environments.

University has been created especially for this thesis and restricts movement to the map of the Kilburn and IT buildings in the University of Manchester shown in Figure 3.3. The purpose of a third map is to test the behaviour of the different neighbour discovery protocols in more densely populated scenarios than either the Manhattan or Helsinki simulations. Previous studies have found that inter-encounter times of humans in office and conference settings follow a power law distribution over the range of 10 minutes to 1 day [Hui, 2008, Wang et al., 2009b]. When paired with the WDM movement model, the maps outlined in Table 3.1 give the same power law distribution [Ekman et al., 2008]. As the University model is new, its inter-encounter distribution when used with the WDM model is presented in Figure 3.4.

In the simulations carried out for this chapter, each map and movement model combination is repeated using different movement speed settings for the movement model. The speed ranges available to simulated participants are either 0.8-1.4 m/s to model human walking speeds, or 7-30 m/s which mimics fast traffic speeds. It is not intended to suggest that PMWDs can be driven around the IT building at the University of Manchester, the different movements speeds are simply used to test the behaviour of the inter-probe time calculation proposed in the next section.

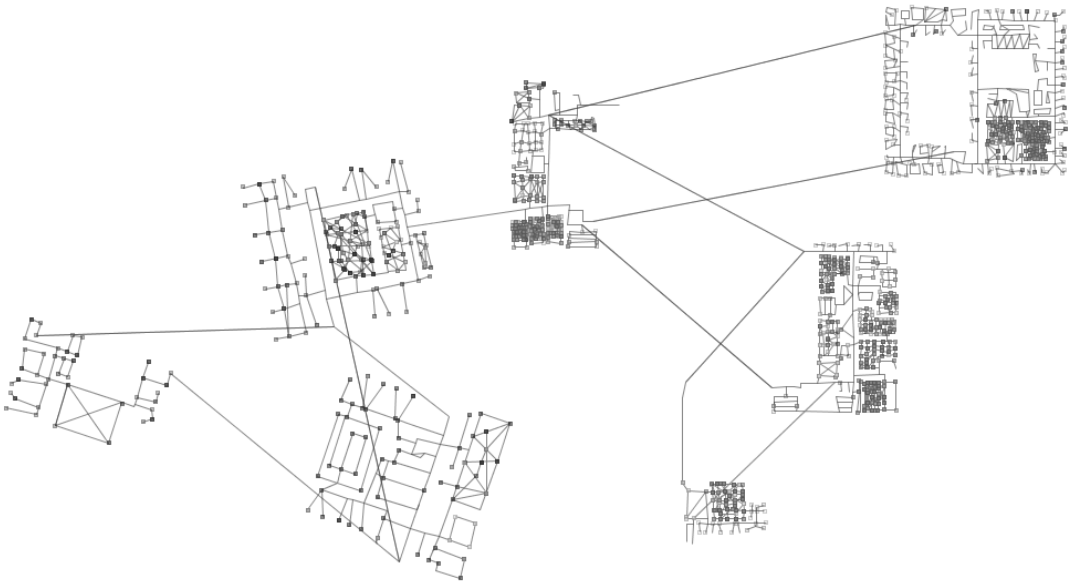


Figure 3.3: The University layout. The lines represent paths that PMWDs can follow in the IT and Kilburn buildings at the University of Manchester. Long lines between sections represent stairs between floors. The dots are points of interest used by the WDM model. They are placed at the locations of offices, lecture theatres, and cafés.

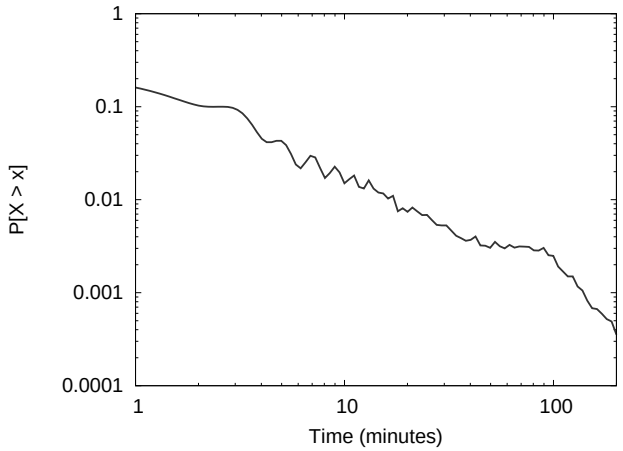



Figure 3.4: Cumulative probability distribution of inter-encounter times between devices in University using the WDM model.

## 3.3 Inter-probe time calculation

Troël stated [Troël, 2004] that a PMWD which comes within a given distance in the transmission radius should be detected as a neighbour (translation from French provided by [Ingelrest et al., 2007]). To facilitate this, Troël proposed Equation 3.1 where $f_{opt}$ is the optimal inter-probe time which depends on the relative movement speed $B$ (which can be detected on most modern mobile phones [Bedogni et al., 2012]) divided by a threshold distance. The threshold distance is calculated using the value $a$ in the range $0 < a \leq 1$ which is multiplied by a maximum uniform transmission radius $R$.

Troël's equation uses $a$ as a means of specifying how close PMWDs should be to each other in relation to the maximum transmission radius before being classified as neighbours. Thus inter-probe times are increased with lower values for $a$ as it is assumed that devices which are within the threshold distance spend more time within transmission range than those outside of the threshold distance.

$$f_{opt} = \frac{2B}{aR} \tag{3.1}$$

Possible issues with Troël's equation include that it assumes that neighbours are devices which are within a threshold distance, and that mobility of neighbours tends toward the centre of the transmission radius. Troël's equation also produces longer inter-probe times when transmission radius decreases. In PSNs, and other networks where devices can move past each other on the outskirts of transmission range, shorter transmission range means that there is less time to communicate with nearby devices. Therefore shorter inter-probe times are needed when transmission range is decreased.

The first contribution of this thesis towards meeting the objectives outlined in Section 1.6.1 is to propose a new equation for calculating inter-probe times for PSNs. The proposed calculation is suitable for any wireless network where devices can move past each other on the outskirts of transmission range. As with Troël's method, the first half of the new equation for Inter-Probe time Calculation (IPC) shown in Equation 3.2 requires a working knowledge of the maximum movement speed at which PMWDs can travel ($B$) measured in metres per second. Equation 3.2 then calculates a sampling rate ($f_s$) using the transmission radius of the device in metres ($R$).

$$f_s = \frac{2B}{R} \tag{3.2}$$

$$\tau = \frac{1}{\alpha f_s} \tag{3.3}$$

The second half of IPC seen in Equation 3.3 produces the inter-probe time referred to as $\tau$. Equation 3.3 also introduces the variable $\alpha$ which should be within the range $0 < \alpha \leq 1$, and is used to specify the required level of connectivity for the network.

Together, Equations 3.2 and 3.3 form the IPC, and unlike Troël's method, produce shorter inter-probe times when transmission range decreases. This is more suited to PSNs, as PMWDs with shorter transmission range may have less time to communicate

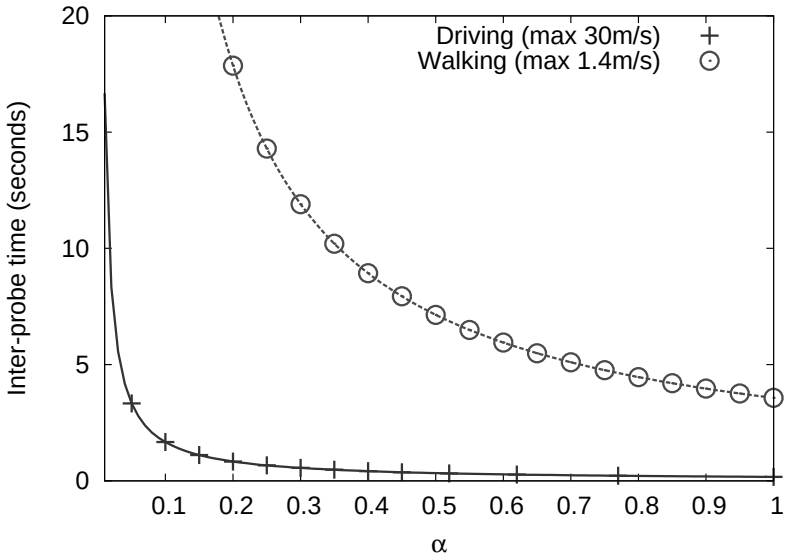



Figure 3.5: Inter-probe times produced by the IPC equation for slow and fast moving PMWDs with a transmission range of 10 meters and $\alpha$ values between 0.01-1.

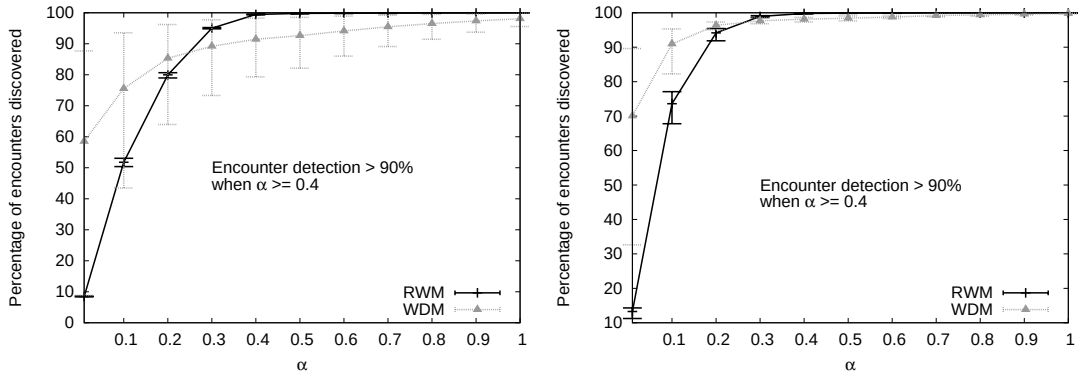
when moving past each other, and must compensate by sending beacons more often.

One important issue to note here is that the faster the movement speed of participants the less time can be given for neighbour discovery intervals with IPC. The length of neighbour discovery intervals must be less than or equal to the inter-probe time. For example, when a movement speed of 30 m/s is used with an $\alpha$ of 1, this leaves 0.166 seconds with which to perform a neighbour discovery interval. Another important issue with the IPC equation is how to choose suitable values for $\alpha$, which will now be discussed in more detail.

### 3.3.1 Choosing $\alpha$

As Section 3.4 will show, a smaller $\alpha$ results in longer inter-probe times, fewer detected encounters, and produces a less connected network. What value $\alpha$ finally takes will be application specific. An $\alpha$ of 1 should only be used for applications where a high degree of connectivity is essential as this will result in more neighbour discovery intervals. In energy constrained environments, or applications where it is permissible to miss some encounters, a longer inter-probe time is desirable and $\alpha$ can be lowered.

To show how $\alpha$ changes the inter-probe times produced by IPC, consider PMWDs moving at a leisurely 1.4 m/s with a transmission range of 10 m. The resulting $\tau$ for these PMWDs using an $\alpha$ of 1 will be 3.57 seconds. If detecting every PMWD in range is not a priority, then $\alpha$ can be lowered. For example, an $\alpha$ of 0.01 using the same figures for movement speed and transmission range will produce a very long inter-probe time of 357.14 seconds.




(a) Slow movement.

(b) Fast movement.

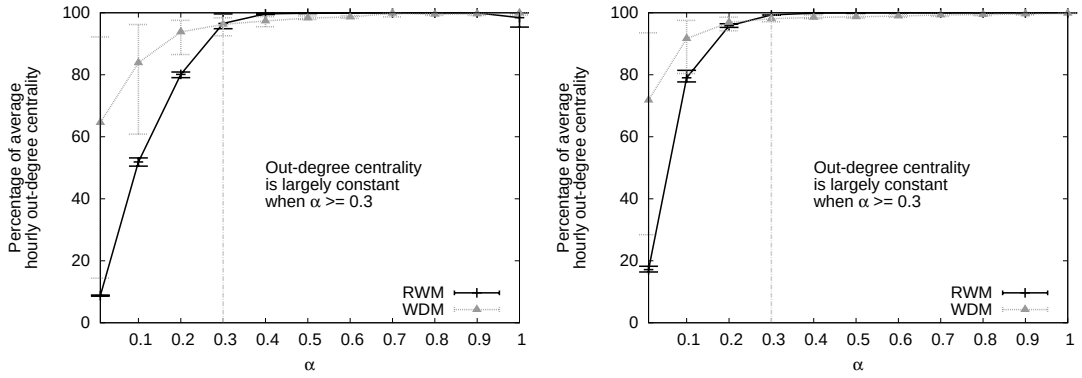
Figure 3.6: Percentage of encounters detected by IPC using $\alpha$ values between 0.01 and 1, with different movement models and speeds.

Further examples of inter-probe times calculated by IPC for cases where the maximum speed of PMWDs is set at 1.4 m/s and 30 m/s are provided in Figure 3.5. The effects of different $\alpha$ values on other network measurements are presented in the following section, starting with the effect on the number of missed encounters.

## 3.4 Neighbour discovery results

Figure 3.6 shows the effect $\alpha$ has on the number of encounters captured using the WDM and RWM models. The figures show the median, minimum, and maximum results taken from the Helsinki, Manhattan, and University experiments that were described in Section 3.2. Encounter percentage is measured against the theoretical best case where PMWDs know as soon as another comes into transmission range.

It is stated in [Keranen et al., 2009] that PMWDs which move to the WDM pattern rather than RWM exhibit more continuous co-location, and as a result simulated PSNs created using WDM are more resilient to longer inter-probe times. Yet Figure 3.6 shows that this claim is not entirely straightforward. Whilst $\alpha$ values below 0.275 result in a higher percentage of encounters detected in the WDM experiments than when using the RWM model, $\alpha$ values above 0.275 actually detect a slightly lower percentage of encounters in the WDM model simulations than in those using the RWM model. This is partly to do with the different number of encounters when using the RWM and WDM models in bounded regions. For example, RWM in the University scenario creates 320,000 possible encounters per day, whereas only 4,000 are created when using WDM in University. Therefore a missed encounter in WDM experiments is more significant than when using the RWM model.




(a) Slow movement.

(b) Fast movement.

Figure 3.7: Hourly out-degree centrality percentage using $\alpha$ values 0.01-1 in IPC for different movement models and speeds.

### 3.4.1 Hourly out-degree centrality

Hourly out-degree centrality is plotted against different $\alpha$ values in Figure 3.7 in order to show how inter-probe time affects network connectivity. Hourly out-degree centrality can be used to measure the number of PMWDs encountered in hourly time frames and therefore has implications for timely data delivery. As encounters in PSNs are asymmetrical [Madan et al., 2010], hourly out-degree centrality measures the number of beacons that a PMWD successfully sends to others, it is a measure of gregariousness of PMWDs.

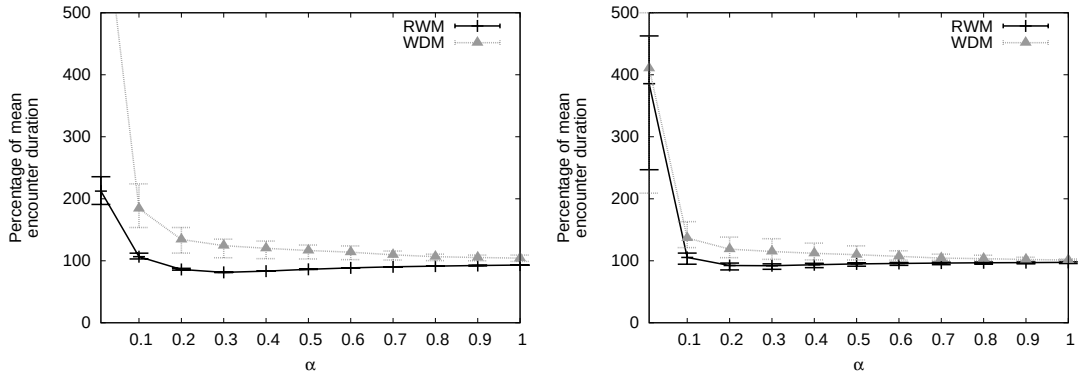
Using the IPC equations from Section 3.3, hourly out-degree centrality follows the same curved shape as the total number of encounters captured in Figure 3.6. Moreover, Figure 3.7 shows that out-degree centrality of PMWDs is largely unaffected when using $\alpha$ values between 0.3 and 1. This information can be used to suggest 0.3 as a suitable default value for PMWDs using IPC, and 0.3 will also be used as a lower bound for $\alpha$ in PISTONS later on in Section 3.5 for this reason.

### 3.4.2 Encounter duration

The total cumulative encounter duration of all PMWDs in the network is also critically dependent on inter-probe times in both of the movement models tested. Figure 3.8 and Figure 3.9 show that the mean encounter duration increases with decreases to $\alpha$, whilst total encounter duration in the network goes down when $\alpha$ is decreased. This indicates that fewer short encounters are being detected when $\alpha$ is decreased, which Figure 3.10a and Figure 3.10b confirms for both the RWM and WDM models.

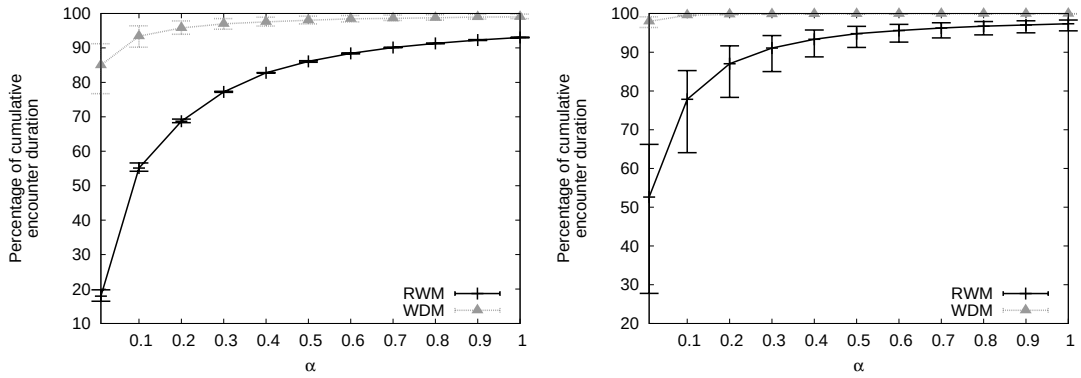
### 3.4.3 Short encounters matter

The results in this chapter show that longer inter-probe times will detect fewer shorter encounters, but what effect does this have on network connectivity overall? Figure 3.11




(a) Slow movement. (b) Fast movement.

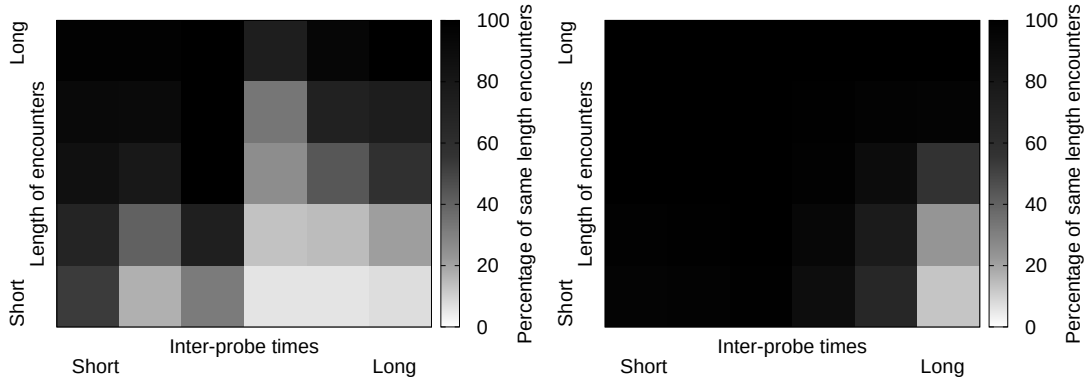
Figure 3.8: Mean encounter duration using different $\alpha$ values compared with the best case neighbour discovery.




(a) Slow movement. (b) Fast movement.

Figure 3.9: Cumulative encounter duration percentage for IPC using different $\alpha$ values compared with the best case neighbour discovery.

shows the relationship between hourly out-degree centrality and shorter encounters for both the WDM and RWM models. Individual encounter duration in the WDM model, and in PSNs, has a heavy tailed distribution [Hui, 2008, Wang et al., 2009b], with the vast majority of encounters being brief. Therefore the loss of short encounters has a drastic effect on network out-degree centrality and network connectivity. Figure 3.11 shows a very high correlation between the number of shorter encounters discovered and out-degree centrality in both the RWM and WDM models. The near linear relationship suggests that network connectivity in PSNs comes from highly mobile PMWDs, rather than the static PMWDs. Put another way, detecting the shorter encounters is essential for preserving network connectivity in PSNs, as the most gregarious PMWDs which connect to the highest number of others, do so for short periods of time.




(a) WDM (b) RWM

Figure 3.10: The numbers of short encounters detected drops as the time between symmetric neighbour discovery intervals (the inter-probe time) is increased.

## 3.5 Asynchronous symmetric neighbour discovery

This section presents a dynamic inter-probe time selection process for PMWDs in PSNs called PISTONS. This new approach alters the inter-probe times of PMWDs based on their movement speed and the results of neighbour discovery intervals. With PISTONS, PMWDs can change their own inter-probe time, and therefore symmetric neighbour discovery intervals are no longer synchronised. However, PMWDs are still only able to detect and reply to neighbour discovery beacons during neighbour discovery intervals.

PISTONS aims to save energy in PSNs by increasing inter-probe times using the process described in Algorithm 1 when a neighbour discovery interval results in no PMWDs detected. PISTONS uses the IPC equations in order to choose inter-probe times, lowering $\alpha$ slightly each time a neighbour discovery interval fails to detect another PMWD. When an encounter between two PMWDs has been detected, $\alpha$ is reset to $\alpha = 1$. This is because a single encounter is treated as an indication that more may follow due to the bursty nature of PSNs [Wang et al., 2009b].

PISTONS does not partition time into arbitrary time frames, or attempt to base inter-probe times on previous encounters because of the possibility of not matching individual scenarios. Furthermore, framing requires complicated stratification as human encounters are self-similar [Wang et al., 2009b] (See all the variations of STAR in [Wang et al., 2009a]). Stratification of time frames may improve the performance of PISTONS in some real world networks, but is costly in terms of implementation and not easy to translate to other movement patterns or device densities.

In PISTONS, inter-probe times are increased by decrementing the variable $\alpha$ from the IPC algorithm in Section 3.3 each time a neighbour discovery attempt is unsuccessful. However, $\alpha$ is never lowered further than 0.3 as the curve of hourly out-degree centrality seen in Section 3.4.1 decreases rapidly at $\alpha < 0.3$, and PISTONS must guard

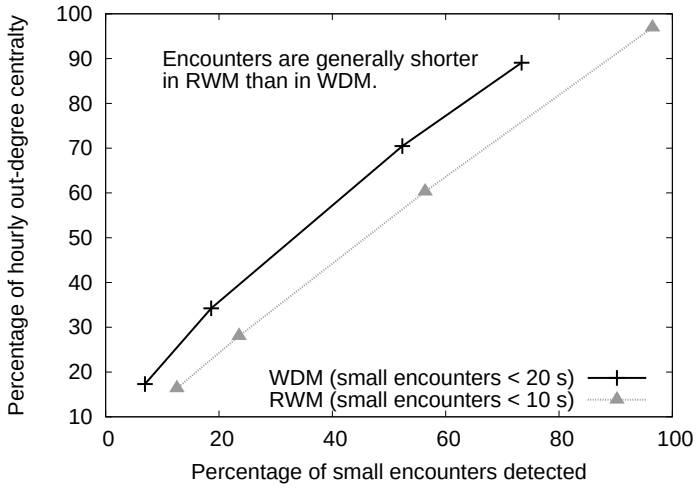



Figure 3.11: Correlation between the number of short term encounters and out-degree centrality.

against the loss of short encounters in order to preserve network connectivity (see Section 3.4.3). For example, if $\alpha$ had a minimum value of 0.01 in University experiments containing fast moving PMWDs moving to the RWM model, then mean hourly out-degree centrality would be 0.581. If the minimum allowed $\alpha$ is set to 0.3 in the same example, then mean hourly out-degree centrality is higher at 0.629.

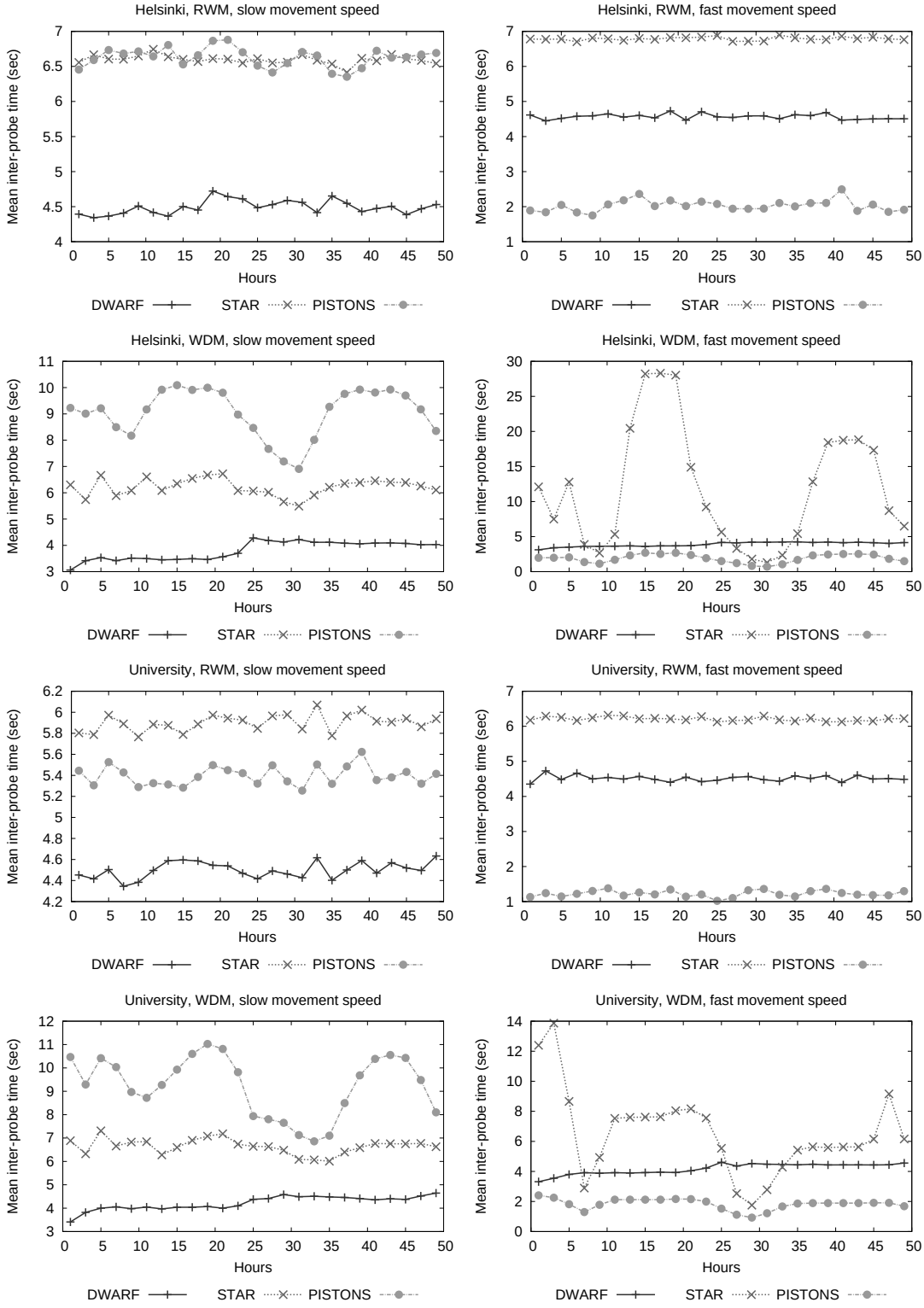
By testing each algorithm using the scenarios and movement speeds presented in Section 3.2, Section 3.5.1 will present a comparison of the asynchronous symmetric neighbour discovery intervals produced by PISTONS to those of two other methods from the literature, STAR and DWARF. Section 3.5.1 will also show that PISTONS can adapt to changing network conditions giving longer average inter-probe times than IPC, whilst preserving mean hourly out-degree centrality considerably better than the other algorithms tested.

### 3.5.1 Results

The inter-probe times produced by PISTONS, STAR, and DWARF for the WDM and RWM models are presented in Figure 3.12. The figures show that PISTONS tends to lower its mean inter-probe time more than the other algorithms when movement speeds increase. It is also interesting to note that the mean inter-probe time of DWARF does not change with changes to movement speed. This is due to the DWARF algorithm provided in [Izumikawa et al., 2010] which does not alter mean inter-probe time when the density of PMWDs in the area around the sensing PMWD does not change.

#### Cumulative encounter duration

Figure 3.13 shows that PISTONS performs as well if not better than DWARF and STAR in terms of preserving cumulative encounter duration in the random movement




(a) Slow movement (b) Fast movement

Figure 3.12: Mean inter-probe times for different movement speeds.

**Algorithm 1** PISTONS algorithm.

```
1:  if CurrentTime > (LastChecked + InterProbeTime) then
2:      Enter neighbour discovery state
3:      if Neighbour Detected then
4:          α = 1
5:          SetInterProbeTime(α)
6:      else
7:          if α > 0.3 then
8:              α = α − 0.01
9:          end if
10:         SetInterProbeTime(α)
11:     end if
12:     LastChecked = CurrentTime
13: end if
14: procedure SetInterProbeTime(α)
15:     f_s = 2B/R
16:     InterProbeTime = τ = (1/f_s)/α
17: end procedure
```

scenarios. This is because PISTONS tends to have a shorter inter-probe time than either DWARF or STAR when movement speed increases, and can therefore detect encounters more quickly. Figure 3.13 also shows that movement speed has little effect on cumulative encounter duration in the WDM model scenarios.

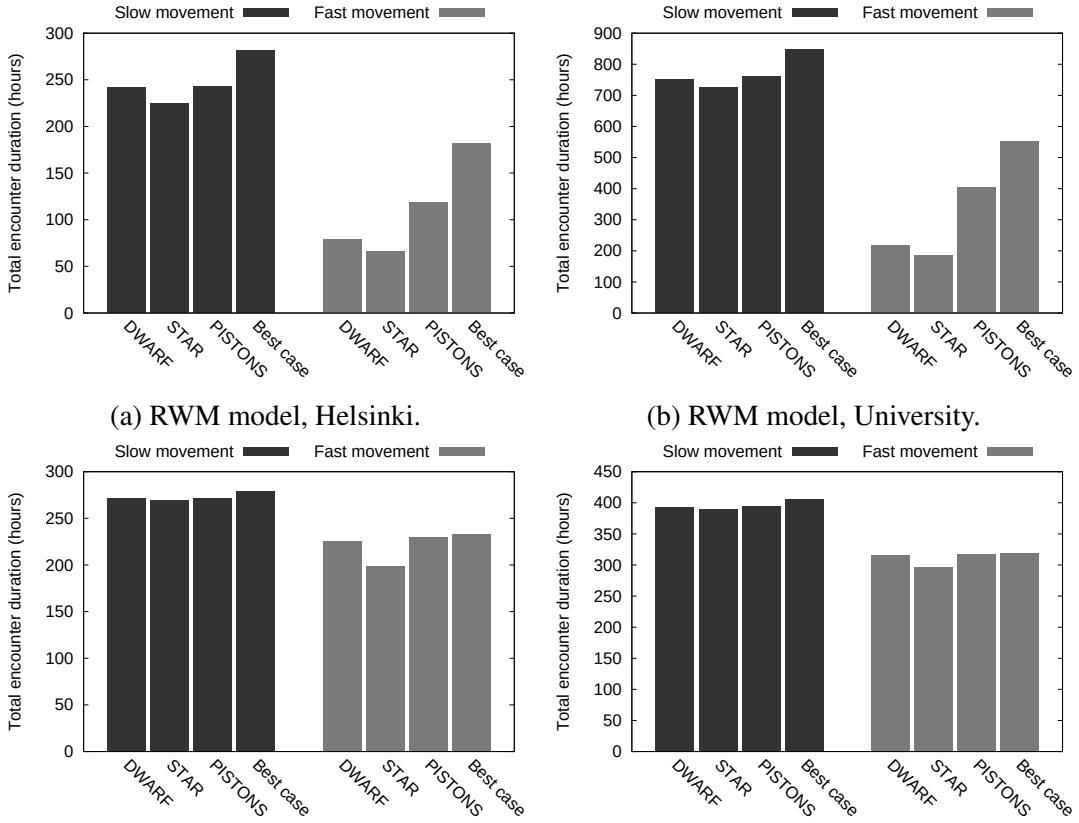
### Out-degree centrality

Figure 3.14 and Figure 3.15 show the resulting mean hourly out-degree centrality from the different algorithms. Generally DWARF preserves as much of the out-degree centrality of PMWDs in slower environments as PISTONS, with PISTONS being better at preserving out-degree centrality when speeds increase because of the IPC equation returning lower inter-probe times than DWARF and STAR (as is shown in Figure 3.12).

Figure 3.14 illustrates the mean hourly out-degree centrality of PMWDs using the RWM model. In these examples, PISTONS handles faster scenarios better than either DWARF or STAR because it detects the shorter encounters with shorter inter-probe times. The outcome of this is that PISTONS only loses 20% of out-degree centrality on average in fast RWM model scenarios, whereas STAR loses 73%.

Despite exploiting self-similar human encounters patterns [Wang et al., 2009b], the mean hourly out-degree centrality of STAR is 55% lower than the best case in the University scenarios using the WDM model. In the same experiments, PISTONS only loses 18% of the mean hourly out-degree centrality compared to the best case, whereas DWARF loses 45%.[1] However, DWARF will actually use a quarter of the neighbour

[1]These figures take into account both slow and fast movement speeds




(a) RWM model, Helsinki.

(b) RWM model, University.

(c) WDM model, Helsinki.

(d) WDM model, University.

Figure 3.13: Total encounter duration for the different experiments compared with the best case that is only possible when PMWDs know as soon as another comes within transmission range.

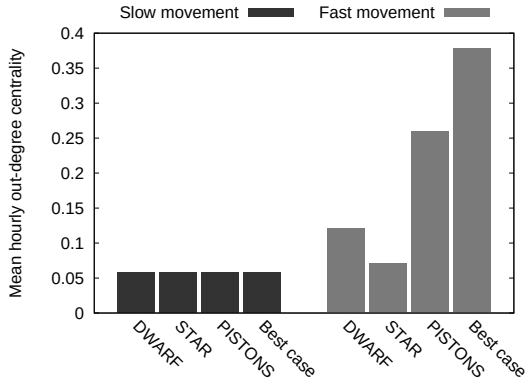
discovery intervals produced by PISTONS and is therefore the most efficient protocol over these scenarios.

## 3.6 Summary

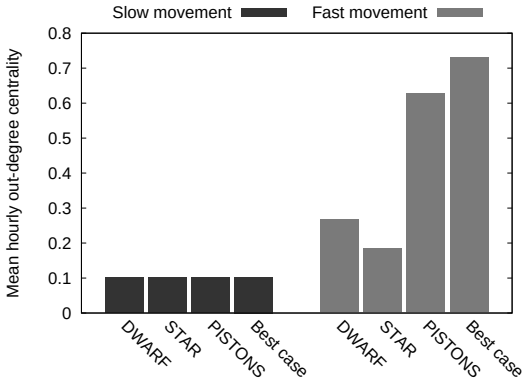
This chapter offers an inter-probe time calculation equation which can be used to meet the objectives outlined in Section 1.6.1, and which can ensure that encounters between PMWDs in PSNs are not missed. Hourly out-degree centrality is used as a measure of temporal connectivity due to the ever changing nature of PSNs. The contributions contained within this chapter included:

1. The IPC method with which to determine suitable inter-probe times for PSNs and other mobile distributed systems.

2. The PISTONS algorithm which can vary inter-probe times to lower the number of symmetric neighbour discovery intervals during quieter periods.

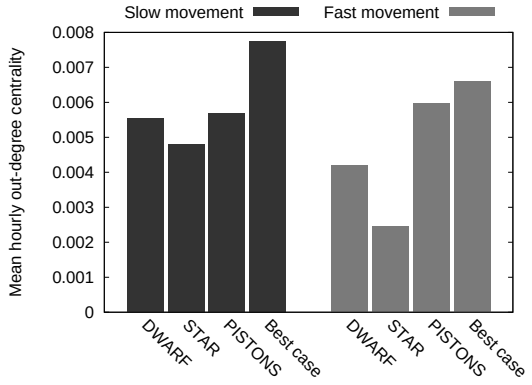



(a) RWM model, Helsinki.

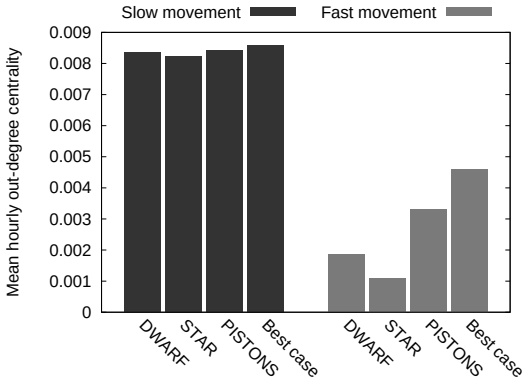



(b) RWM model, University.

Figure 3.14: Mean hourly out-degree centrality for PISTONS, STAR, and DWARF in RWM model scenarios compared with the best case.




(a) WDM model, Helsinki.




(b) WDM model, University.

Figure 3.15: Mean hourly out-degree centrality for PISTONS, STAR, and DWARF in WDM model scenarios compared with the best case.

3. Discovery of a correlation between the number of short term encounters in a PSN and out-degree centrality.

4. Discovery that devices moving to the WDM model are not always more resilient to longer inter-probe times than devices moving to the RWM model.

In summary, short inter-probe times are needed in order to detect short encounters (encounters which last less than 20 seconds) and to maximise network connectivity in PSNs. An inter-probe time of 3.57 seconds for PSNs made entirely of pedestrians would be sufficient to discover over 99% of encounters due to the limited speed and patterns of movement of the participants (see Section 3.3).

However, such a short time between neighbour discovery intervals may not be needed between the bursts of encounters described in Section 2.4. Section 3.5 introduced a novel asynchronous symmetric neighbour discovery protocol called PISTONS

with which PMWDs can increase their own inter-probe time during periods of inactivity whilst still detecting the majority of short encounters.

Compared with IPC using $\alpha = 1$, PISTONS reduces the number of inquiry intervals needed in WDM experiments by 65%, but this incurs a small 2.7% loss in hourly out-degree centrality. Whether this 2.7% loss in out-degree centrality is critical for data dissemination applications in PSNs remains to be seen. This may depend on a number of other factors not explored here including how are the shortest and fastest paths between PMWDs affected by IPC and PISTONS?

As an interesting side note, if the inter-probe times generated by IPC are compared to the ones used in current reality mining experiments shown in Table 1.1, then it can be assumed within reasonable doubt that many encounters were missed, and that proximity between wireless interfaces was not being accurately reported. The shortest inter-probe time seen in the reality mining experiments described in Chapter 1 is 60 seconds. To get this value using IPC with a walking speed of 1.4 m/s and transmission range of 10 m gives an out-degree centrality loss prediction of over 99%, which means that the number of encounters between PMWDs would be considerably higher if inter-probe times were lowered in empirical experiments.

# Chapter 4

# Distributed cluster detection

Chapter 3 describes a neighbour discovery processes with which PMWDs can detect encounters with other nearby PMWDs. The next four chapters will describe how the information collected about encounters can be used for efficient opportunistic message delivery in PSNs.

Opportunistic message delivery protocols are preferred in PSNs to the route discovery protocols used in MANETs because it is not always possible to discover stable end-to-end paths between PMWDs. Moreover, opportunistic message delivery protocols which duplicate messages are often used to increase delivery probability at the expense of efficiency.

Rather than obliviously copying messages to every encountered PMWD, strategies which help to keep the number of message copies to a minimum can be used in order to save space, bandwidth, and ultimately energy (see Section 2.3).

It has been demonstrated previously that segregating PSNs into logical partitions known as clusters can significantly lower the number of duplicates needed in order for a message to reach its final destination [Hui and Crowcroft, 2007]. However, considering the many possible clusters which can be created by automatic means (this point will be elaborated upon in Section 4.3), it is important to be able to predict if the cluster partitioning being used is beneficial for message delivery. Using the reality mining datasets from the CRAWDAD repository, the aim of this chapter is to separate the "good" clustering from the "bad" in the context of cluster based opportunistic message delivery.

## 4.1 Distributed clustering characteristics

Section 2.7.1 gave an introduction to some distributed clustering algorithms which can be used within PSNs. This section will describe the clusters generated by distributed clustering algorithms in a more formal way so that we may later understand how cluster partitioning effects cluster based message delivery in PSNs, starting with the definition of local clusters and the minimum contents of local clusters:

**Distributed cluster characteristic 1.** *Assuming there are n PMWDs in the PSN running a distributed cluster detection algorithm, each PMWD creates a mapping from itself to a local cluster set which contains itself and up to* $n-1$ *other PMWDs.*

Whilst it may be argued that local cluster sets need not contain the local PMWD, for simplicity the term local cluster is used instead of local cluster set when describing distributed cluster detection algorithms, and it would be counter intuitive to form local clusters which do not contain the local PMWD. Requiring that PMWDs belong to their own local cluster set (hereby referred to simply as the local cluster) also means that the rule for the maximum number of local clusters produced is as follows.

**Distributed cluster characteristic 2.** *n PMWDs running a distributed cluster detection algorithm will produce n, sometimes identical, non-empty local clusters. Each one of the produced local clusters (collectively referred to as P) being one of the* $2^n - 1$ *possible local clusters.*

As PMWDs are never guaranteed an encounter with another PMWD, they may not have the opportunity to form local clusters larger than a singleton containing themselves. Each local cluster contains at least the ID of the PMWD it has been created by, and possibly any other ID from the $n-1$ other PMWDs. The cardinality $2^n - 1$ of the possible local clusters comes from the cardinality of the power set of PMWDs in the PSN, minus the empty set as no local clusters can be empty.

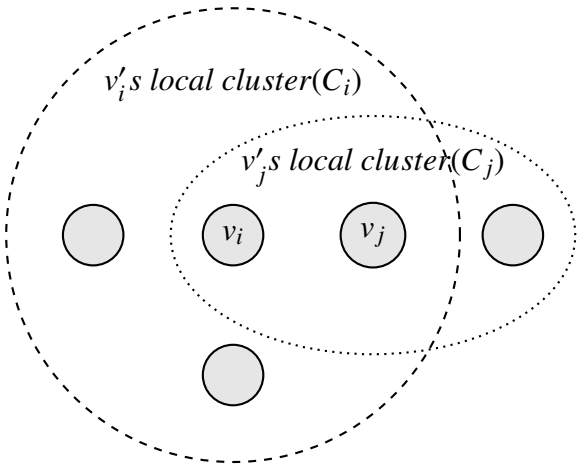



Figure 4.1: Example of local clusters belonging to $v_i$ and $v_j$. Note that when local clusters contain more than one PMWD the partitioning of the local clusters is fuzzy.

**Distributed cluster characteristic 3.** *If the cardinality of a single local cluster* $C_i$ *is* $> 1$*, then* $C_i$ *will overlap with (or completely envelop)* $|C_i| - 1$ *other local clusters.*

**Distributed cluster characteristic 4.** *Collectively, the local clusters from n different PMWDs are a multiset of local clusters with a cardinality of n.*

Some local clusters may be the same as others, and from the collection of local clusters all of the distinct local clusters can be extracted, leaving only the set of local clusters whose members are collectively called the *natural clusters*.

**Distributed cluster characteristic 5.** *The set of local clusters from a PSN is called the set of natural clusters (N):*

$$N = \{x : x \text{ is a local cluster in } P\} \tag{4.1}$$

If all of the $n$ local clusters are different, then $|N| = n$. Some other properties of natural clusters to note are: Every PMWD is guaranteed to be in at least one natural cluster, and natural clusters can be subsets of other natural clusters.

## 4.2 Opportunistic message delivery using clusters

Most ad hoc routing protocols for MANETs work with a single copy of a message as they do not need to duplicate messages in ways which might increase delivery probability [Perkins and Royer, 1999, Jacquet et al., 2001]. In contrast, a common characteristic which many opportunistic message delivery protocols share is the duplication of messages in ways which increase delivery probability in highly dynamic networks.

The opportunistic message delivery protocols proposed in this thesis all use clusters of closely related PMWDs as boundaries within which to copy messages. The clusters help to stop excessive message duplication, meaning that cluster based opportunistic message delivery is often significantly more efficient than oblivious flooding techniques such as Epidemic [Hui and Crowcroft, 2007].

Distributed clustering algorithms such as those described in Section 2.7.1 can be used to partition PMWDs into clusters, but they do not dictate how clusters should be used to deliver messages. Cluster based opportunistic message delivery can therefore be tackled separately. The objective of Section 4.3 is to explore the suitability of different cluster partitions for opportunistic message delivery in PSNs. In order to meet this aim, Algorithm 2 will be used to test the message delivery properties of different cluster partitions. This multiple copy opportunistic message delivery algorithm simply floods clusters containing the final destination with as many copies of a message as possible whilst attempting to deliver a message.

### 4.2.1 Experimental environment

The opportunistic message delivery experiments in this thesis are all conducted using the same encounter information obtained during reality mining experiments, and by using the ONE simulator to create virtual wireless data connections between PMWDs. Despite encounters in reality mining datasets being asymmetrical, in this analysis encounters are being used to represent “data transfer opportunities that each of the participants would have, if they were equipped with [PMWDs] which are always on and

**Algorithm 2** Cluster based message delivery
The local cluster of a particular PMWD $v_i$ is denoted $c_i$.
**input:**
A list of all connected PMWDs to $v_i$, $E_i$.
A message called $m$.
The final destination of $m$ is $v_j$.

```
for all E_i as v_x do
    if v_x = v_j then
        DeliverMessage(m)
    else
        if v_j ∈ c_x then
            CopyMessageToEncounteredPMWD(m, v_x)
        end if
    end if
end for
```

always carried" [Scott et al., 2009]. This means that simulations approximate the conditions found in potential future PSNs. However, one drawback to this approach is that the simulated PSNs are fragmented due to limited participation, short wireless transmission range, and large geographical areas.

The three reality mining datasets that are used for opportunistic message delivery experiments in Sections 4.3 and 4.4 are shown in Table 4.1. They are an experiment from LocShare labelled UCL1 [Abdesslem et al., 2011], and Infocom5 and Cambridge from the Haggle project [Scott et al., 2009]. In each case the external and static devices present in the datasets have been removed to concentrate on clusters formed solely by encounters between PMWDs.

In every opportunistic message delivery experiment conducted for this thesis, a new 1 KB message is generated every 30 seconds by one random PMWD to send to another, encounters between PMWDs represent 250 kBps each way data connections, and every PMWD has a 5 MB message buffer.

The results in Table 4.1 are compiled from five separate experiments on each dataset so that many different paths through the PSN are explored. The Time To Live (TTL) of messages has been set to 1 hour in order to give an idea of the timescales needed to deliver messages. Once the TTL for a message has expired then the message is deleted. In the worst case referred to as *Wait* there is no attempt at message forwarding. When using Wait messages are only delivered directly to the final recipients if an encounter between the source and destination takes place within the TTL.

The overheads associated with Epidemic message delivery shown in Table 4.1 were calculated using the number of messages relayed, minus the number of messages delivered within the TTL, divided by the number of messages delivered within the TTL. Whilst not accounting for all costs associated with the wireless medium, this measure is sufficient for this study, as lower overheads imply fewer duplicate messages.

| | UCL1 | Infocom5 | Cambridge |
|---|---|---|---|
| **Duration (days)** | 6 | 3 | 12 |
| **PMWDs** | 20 | 41 | 36 |
| **Number of encounters** | 512 | 28216 | 21239 |
| **Message TTL** | 1 hour | | |
| **Message generation frequency** | 120 messages per hour | | |
| **Transmit speed** | 250 kBps | | |
| **Message size** | 1 KB | | |
| **Buffer size** | 5 MB | | |
| **Delivery probability of Epidemic** | 0.012 | 0.201 | 0.079 |
| **Delivery probability of Wait** | 0.010 | 0.080 | 0.030 |
| **Epidemic overheads** | 9.623 | 37.601 | 32.732 |

Table 4.1: Opportunistic message delivery performance of Epidemic and Wait in some reality mining datasets using the ONE simulator.

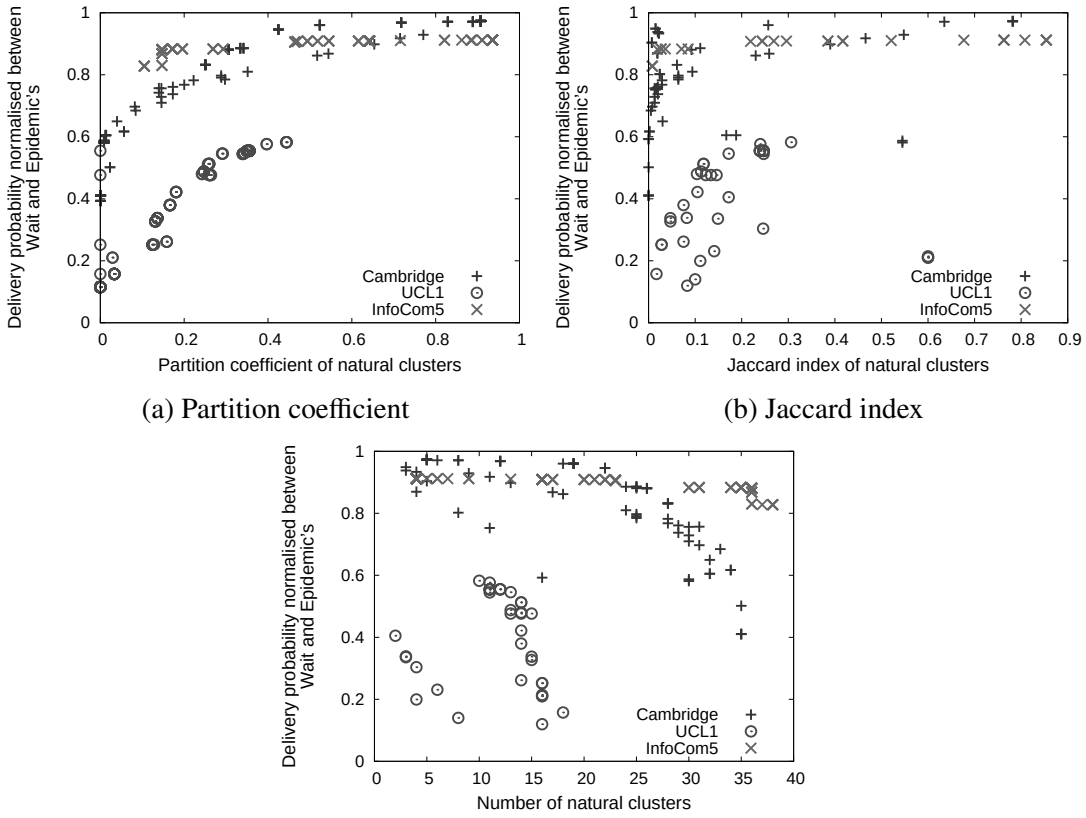
Table 4.1 also shows that message delivery probability within 1 hour is extremely low even when considering the "best case" achieved using Epidemic. If the TTL restriction are relaxed to allow infinite TTL then the best case delivery probability achieved with Epidemic increases for UCL1, Infocom5, and Cambridge to 0.61, 0.92 and 0.98 respectively. In order for opportunistic message delivery to be timely and reliable in PSNs, global PMWD density as a function of transmission range would need to be high enough so that a path exists between each PMWD at all times. However, low numbers of participants and short transmission range of participants mean that delivering every message without loss is impossible in these datasets.

## 4.3 Analysing the clusters used in message delivery

The terminology introduced in Section 4.1 can be used to analyse the clusters produced by distributed algorithms in the reality mining datasets. The number, size, and membership of these clusters relies heavily on the algorithm used and the values of any parameters specified by the user. This section investigates how the "quality" of cluster partitioning affects message delivery probability in PSNs using the message delivery algorithm outlined in Algorithm 2 and the cluster analysis techniques discussed in Section 2.7.2.

The clusters used in this analysis were generated using the Simple and $k$-Clique distributed clustering algorithms.[1] In order to explore as many different cluster partitions as possible, the same five random message generation patterns that were used

[1]Special thanks to P.J. Dillon at the University of Pittsburgh who provided the code for the ONE Simulator to generate Simple and $k$-Clique clusters.




(a) Partition coefficient

(b) Jaccard index

(c) Number of natural clusters

Figure 4.2: Analysis of clusters detected using the Simple and $k$-Clique distributed clustering algorithms, and message delivery probability.

on each dataset for the results in Table 4.1 have been repeated over a large range of parameters for each clustering algorithm. For example, $\lambda$ values in the range of 0.01 to 0.99 are used with Simple along with different values for the merging threshold in the range 0.1 to 0.9 (See [Hui et al., 2007b] for a full explanation of Simple and its other parameters). When generating $k$-Clique clusters, values for $k$ between 3 and 7 have been used.

### 4.3.1 Fuzzy distributed clusters

If there is more than one PMWD running a distributed clustering algorithm, then there is a possibility that a single PMWD can belong to more than one natural cluster. In other words, as $n$ different PMWDs can independently determine their own local clusters, a single PMWD can belong to $n$ different "fuzzy" natural clusters. The fuzziness of the natural clusters produced using Simple and $k$-Clique has been measured using Dunn's partition coefficient [Bezdek, 1981, Trauwaert, 1988] and is presented in Figure 4.2a.

| | UCL1 | Cambridge | Infocom5 |
|---|---|---|---|
| **Partition coefficient** | 0.95 | 0.88 | 0.80 |
| **Jaccard index** | 0.03 | 0.37 | 0.68 |
| **Number of natural clusters** | -0.05 | -0.75 | -0.77 |

Table 4.2: Pearson Correlation Coefficient (PCC) of cluster measurements against message delivery probability.

Figure 4.2a shows that natural clusters that have a greater partition coefficient (and are therefore less fuzzy) give the greatest probability of delivering a message with a Pearson Correlation Coefficient (PCC) shown in Table 4.2 consistently higher than 0.8. Figure 4.2a also shows that message delivery probability approaches that of Epidemic's when the partition coefficient of the clustering is greater than 0.4 in both the Cambridge and Infocom5 datasets.

The reason why delivery probability grows with the partition coefficient is that the partition coefficient generally grows as more PMWDs are added to local clusters. Larger local clusters also increase the likelihood that a message destination will be present in the local cluster of an encountered device.

### 4.3.2 Jaccard index

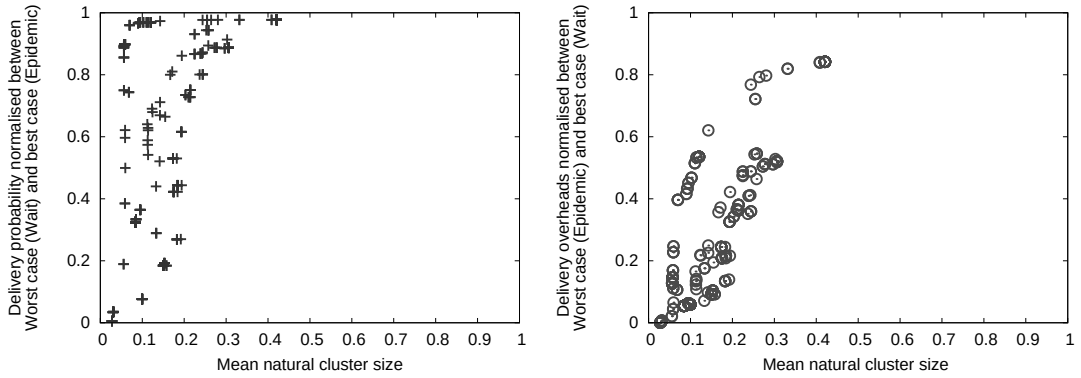
To see the extent that variance between local clusters affects message delivery, the Jaccard index of the natural clusters has been plotted against message delivery probability in Figure 4.2b. Here, the Jaccard index assesses the similarity between the natural clusters reported by the clustering algorithms at the end of each experiment.

Figure 4.2b illustrates that generally the Jaccard index rises with message delivery probability. This is consistent with the analysis done using the partition coefficient, as the Jaccard index will be higher when there are few natural clusters, and the union of their members is similar to their intersection. However, Table 4.2 shows that the correlation of delivery probability with Jaccard index is not as strong as with the partition coefficient.

### 4.3.3 Number of natural clusters

This subsection will briefly discuss the number of natural clusters present at the end of simulations in order to further investigate how PMWDs being added to local clusters can affect cluster based opportunistic message delivery. Table 4.2 and Figure 4.2c show a negative relationship between the number of natural clusters and successful message delivery. In other words, the fewer natural clusters there are, the greater the message delivery probability should be.

The implications of the findings from this and the previous two subsections are that PMWDs should have the ability to lower the number of natural clusters if a high message delivery probability is required. Adding PMWDs to local clusters and merging




(a) Delivery probability (b) Delivery overheads

Figure 4.3: Delivery probability and overheads based on natural cluster size.

local clusters can lower the number of natural clusters in the PSN. However, if all local clusters are merged to create a single natural cluster, then messages will be flooded between PMWDs in the same way as Epidemic. Conversely, if a more efficient but less reliable message delivery performance is required, then the number of natural clusters should be increased by, for example, removing PMWDs from local clusters (as will be done in chapters 6 and 7).

### 4.3.4 Cluster size

The size of natural clusters is closely related to message delivery probability using the message delivery mechanism from Section 4.2. Figure 4.3a shows message delivery probability rising with natural cluster size, and shows a delivery probability close to that of Epidemic's is achieved when mean natural cluster size is roughly a third of that of the dataset. This is intuitive, as the larger the cluster, the more likely it is to contain the final destination.

The flip side to increased message delivery probability using large clusters is that the number of duplicate messages also increases. Figure 4.3b shows that for the same experiments in Figure 4.3a, the overheads tend to increase with natural cluster size. However, overheads are 15% lower than Epidemic's when using clusters that are 33% of size of the dataset size, for only a small decrease in the number of messages delivered.

## 4.4 New distributed clustering algorithms

The findings from the previous section suggested that for effective and efficient message delivery using clusters, distributed clustering algorithms must:

1. Lower the number of natural clusters,

**Algorithm 3** Promote-2
**input:**
User defined cluster merging threshold, $\gamma$.
Local cluster, $c_i$.
Remote PMWD, $v_j$.
Local cluster of $v_j$, $\mathrm{c}_j$.
The familiar device of $v_i$, $p$.

**if** $v_j \notin c_i \wedge v_j = p$ **then**
  $c_i = c_i \cup \{v_j\}$
**end if**
**if** $(|c_i \cap c_j| > (\gamma * |c_i \cup c_j|))$ **then**
  $c_i = c_i \cup c_j$
**end if**

2. Keep clusters to a maximum size of 33% of the dataset size in order to ensure high data-delivery probabilities and provide some efficiency.

To demonstrate how the number of natural clusters can be controlled using a distributed algorithm, two new distributed clustering protocols called Promote-2 and Nomads are proposed in this section. However, neither Promote-2 or Nomads attempt to keep cluster size at 33% of the size of the dataset. This would require PMWDs to know exactly how many other PMWDs are in the PSN which may not be realistic. Moreover, none of the other distributed clustering algorithms proposed in this thesis will attempt to limit cluster size for the same reason, despite the possibility that limiting cluster size may help to improve the efficiency of cluster based opportunistic message delivery. A more detailed discussion on the challenges involved and the possibilities of limiting cluster size can be found later in Section 8.2.4.

### 4.4.1 Promote-2

If familiar thresholds in Simple and $k$-Clique are set too high, then PMWDs with the highest cumulative encounter duration to each other are not guaranteed to be included in their respective local clusters. Promote-2 is a derivative of Simple which guarantees local cluster membership for *familiar devices* and only needs a single parameter which is used to control cluster merging.

In Promote-2, PMWDs keep a count of the cumulative encounter duration with other PMWDs in order to separate the social PMWDs from the non-social ones. If a PMWD $v_j$ has the highest cumulative encounter duration recorded on another PMWD $v_i$, then $v_j$ is the familiar device of $v_i$, and $v_j$ is promoted to the local cluster of $v_i$.

Local clusters are merged in Promote-2 using $\gamma$ in a similar way to how it is used by Simple. However, Promote-2 aims to reduce the number of natural clusters by allowing local clusters to be merged without first checking if the encountered PMWD is in the local cluster. In Promote-2, the local clusters of two PMWDs are always merged if

**Algorithm 4** Nomads cluster formation
**input:**
User defined familiar threshold, $\upsilon$.
User defined cluster merging threshold, $\gamma$.
Local cluster of $v_i$, $c_i$.
Remote PMWD, $v_j$.
Local cluster of $v_j$, $c_j$.
Cumulative encounter duration between $v_i$ and $v_j$, $\delta$.
The familiar device of $v_i$, $p$.

```
if (v_j ∉ c_i ∧ v_j = p) then
    c_i = c_i ∪ {v_j}
end if
if (|c_i ∩ c_j| > (γ ∗ |c_i ∪ c_j|)) then
    c_i = c_i ∪ c_j
end if
if δ ≥ υ then
    if (|c_i ∩ c_j| = 0) then
        c_i = c_i ∪ {v_j}
    end if
end if
```

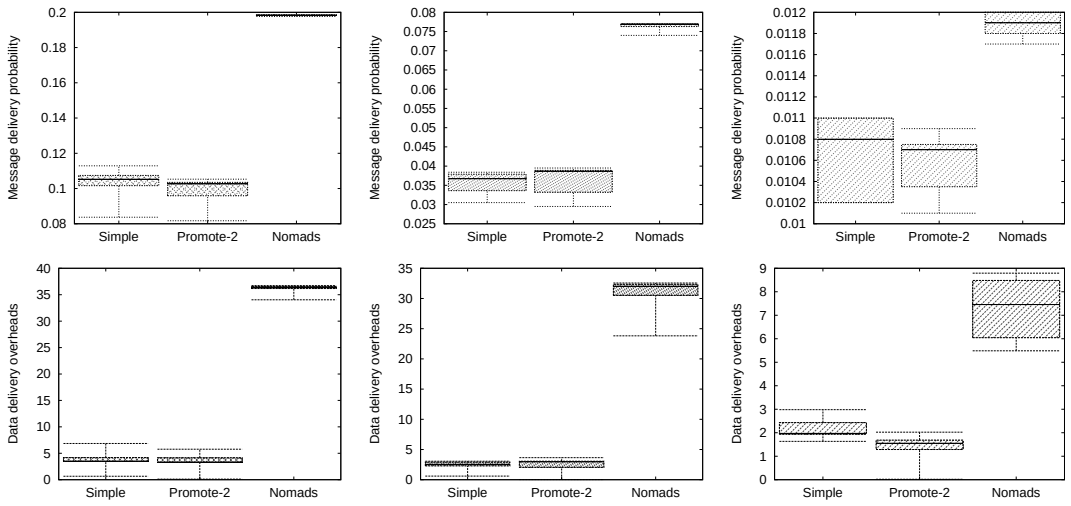
the intersection of the two local clusters is greater than the union multiplied by the user supplied variable $\gamma$, where $0 \leq \gamma \leq 1$. This means that low $\gamma$ values will causes local clusters to merge even if the intersection of local clusters is small compared to the union.

### 4.4.2 Nomads

Nomads as shown in Algorithm 4 allows local clusters to gradually increase in size by adding familiar devices in a similar way to Promote-2. In addition to the operations undertaken by Promote-2, Nomads also adds to local clusters PMWDs that may be wandering between already defined clusters. Nomads does this by checking if the intersection between the local device's local cluster and an encountered device's local clusters is zero. This additional operation ensures that *nomads* which have wandered from remote parts of the network can be included in the local clusters of others. The familiar threshold is reintroduced in the nomads check to stop brief encounters between PMWDs resulting in clusters growing prematurely.

### 4.4.3 Message delivery using Promote-2 and Nomad clusters

Figure 4.4 shows the performance of the opportunistic message delivery algorithm described in Section 4.2 using clusters provided by Simple, Promote-2 and Nomads. As well as the five random message generation patterns described in Section 4.2.1,




(a) Infocom5 (b) Cambridge (c) UCL1

Figure 4.4: Minimum, $1^{st}$ quartile, median, $3^{rd}$ quartile, and max message delivery probabilities and overheads for Simple, Promote-2, and Nomads clusters.

each clustering algorithm has been used with $\gamma$ values between 0.01 and 0.99 with a resolution of one hundredth of a second to create the output for Figure 4.4.

Another important point to make about the simulations for these experiments is that the familiar threshold for the UCL1 experiments had to be lowered compared to the other datasets to account for the fact that encounters in the dataset are inferred, and each one lasts for precisely 10 seconds. Therefore, familiar thresholds for Simple and Nomads have been set at 5000 seconds in the Infocom5 and Cambridge datasets, and 50 seconds in the UCL1 experiments.

Figure 4.4 shows that Nomads consistently delivers more data within the 1 hour TTL than the other algorithms, albeit at a higher cost in terms of overheads. This is because Nomads creates larger local clusters irrespective of the values chosen for user controlled variables. Nomads does this by including PMWDs with no close relationship to each other in local clusters using the test to see if the intersection of two local clusters is zero.

## 4.5 Summary

The findings in this chapter have gone part way to describing suitable clusters for message delivery in the chosen scenarios. The results have indicated that message delivery performance is not reliant on any single characteristic of the local clusters. However, there are a number of known factors which can determine local and natural cluster's ability to influence message forwarding in a PSN:

1. Natural clusters should be strict with a partition coefficient greater than 0.4 for successful message delivery (Section 4.3.1).

2. More natural clusters can adversely affect message delivery probability (Section 4.3.3).

3. Natural cluster size is strongly correlated with message delivery probability and delivery overheads.

An open problem with multiple copy cluster based message delivery is how to stop overheads increasing with cluster size (see Figure 4.4). This issue has been attributed to messages being transferred to obsolete cluster members in the past [Hui et al., 2007b]. Therefore, the temporal issues surrounding cluster membership is looked at in detail in the following three chapters in an attempt to eradicate obsolete cluster membership and lower message delivery overheads.

# Chapter 5

# Spatio-temporal cluster detection

Hui et al. stated that clusters may only be valid for a particular time frame because of the changing movement patterns of humans [Hui, 2008]. Zyba et al. also observed that PMWDs can change their role over time, flipping between social and non-social roles [Zyba et al., 2011]. By exploring these two insights in more detail it may be possible to lower the amount of overheads associated with cluster based opportunistic message delivery whilst still keeping delivery probability high. This chapter introduces a novel way with which to detect clusters in reality mining datasets, one which ensures that clusters are relevant to the period of time in which they are detected.

## 5.1 Aggregated monotonic clustering

The distributed clustering algorithms Simple, Promote-2, and Nomads discussed in Chapter 4 can be used to generate local clusters in PSNs. Critically, each of the algorithms use all of the encounters that PMWDs have detected, with no regard to the time that has passed since the encounter took place; this is called *aggregated clustering*. Furthermore, Simple, Promote-2, and Nomads only add PMWDs to local clusters, they do not remove PMWDs when encounter patterns change and cluster membership becomes obsolete. The end result is clusters that continuously grow in size, which are referred to as *aggregated monotonic clusters*.

It is shown in Section 4.3.4 that aggregated monotonic clustering can limit the size of the spanning tree required by opportunistic message delivery algorithms to deliver a message. Chapter 4 also showed that by using aggregated monotonic clusters as a guide, opportunistic message delivery algorithms which flood the network can almost match the delivery probability of Epidemic but with 15% fewer overheads. However, 15% fewer overheads may still be too inefficient for some applications. Figure 5.1 illustrates the relationship in the Cambridge dataset between cluster size and the number of duplicate messages transferred using Nomads. By the end of this example twenty five duplicate messages are being transferred for every message successfully delivered because of the aggregated monotonic clusters of unbounded size that Nomads produces.

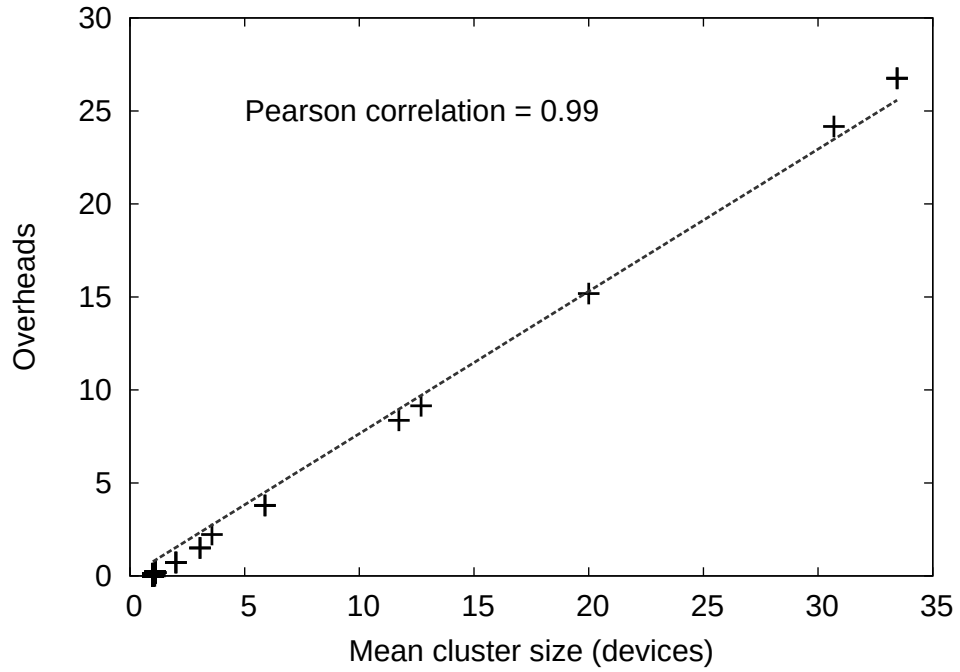



Figure 5.1: Overheads of message delivery compared with aggregated monotonic cluster size for the Nomads algorithm in the Cambridge dataset.

Unfortunately, substituting the distributed clustering algorithm in Nomads for one which has bounded cluster size (and should therefore limit overheads) is not trivial. Budget-based distributed clustering algorithms that attempt to limit cluster size can still produce very large clusters in dense regions of the network [Abdesslem et al., 2007, Krishnan and Starobinski, 2006], and simply limiting the size of clusters will not stop obsolete membership from persisting.

## 5.2 Spatio-temporal clustering

A different approach to aggregated monotonic and budget-based clustering is required to ensure that clusters do not contain obsolete members. *Spatio-temporal clustering* is a multi-dimensional technique with which to partition time series data, and which ensures that clusters have some relevance to temporal regions in the data [Kelso, 1988]. To better understand the difference between spatio-temporal and aggregated monotonic cluster detection, let us consider the Cambridge example from Figure 5.1 once more. Figure 5.2 shows all of the encounters between the 36 PMWDs in the Cambridge reality mining dataset. One possible aggregated monotonic cluster of this data is all of the PMWDs with edges incident upon them, i.e. all of the PMWDs that have been encountered by others. However, this tells us nothing about the start time or duration of individual encounters.

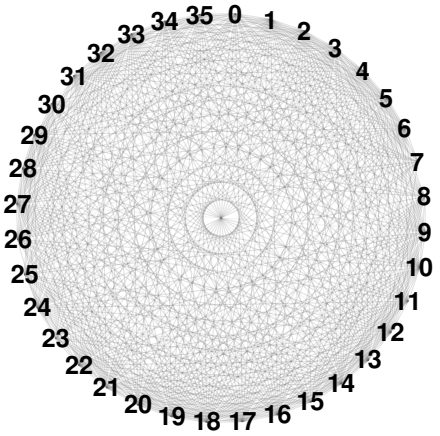
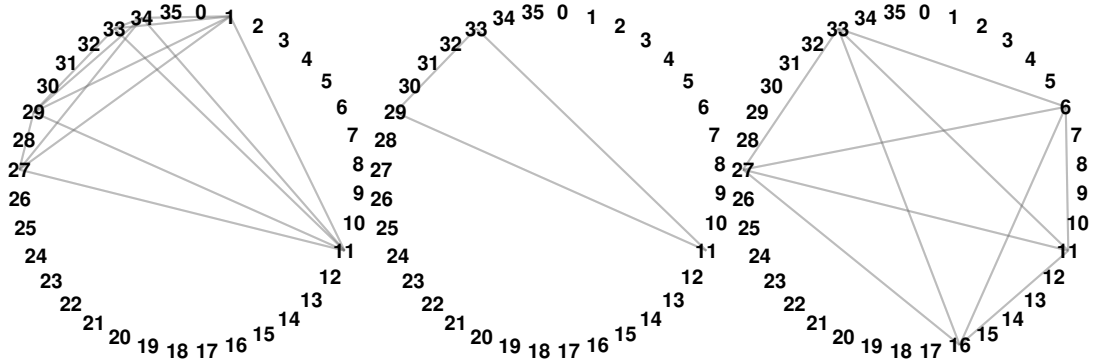


Figure 5.2: Aggregated encounters in the Cambridge dataset. Almost all of the information from the experiment is lost when data is presented in this way.

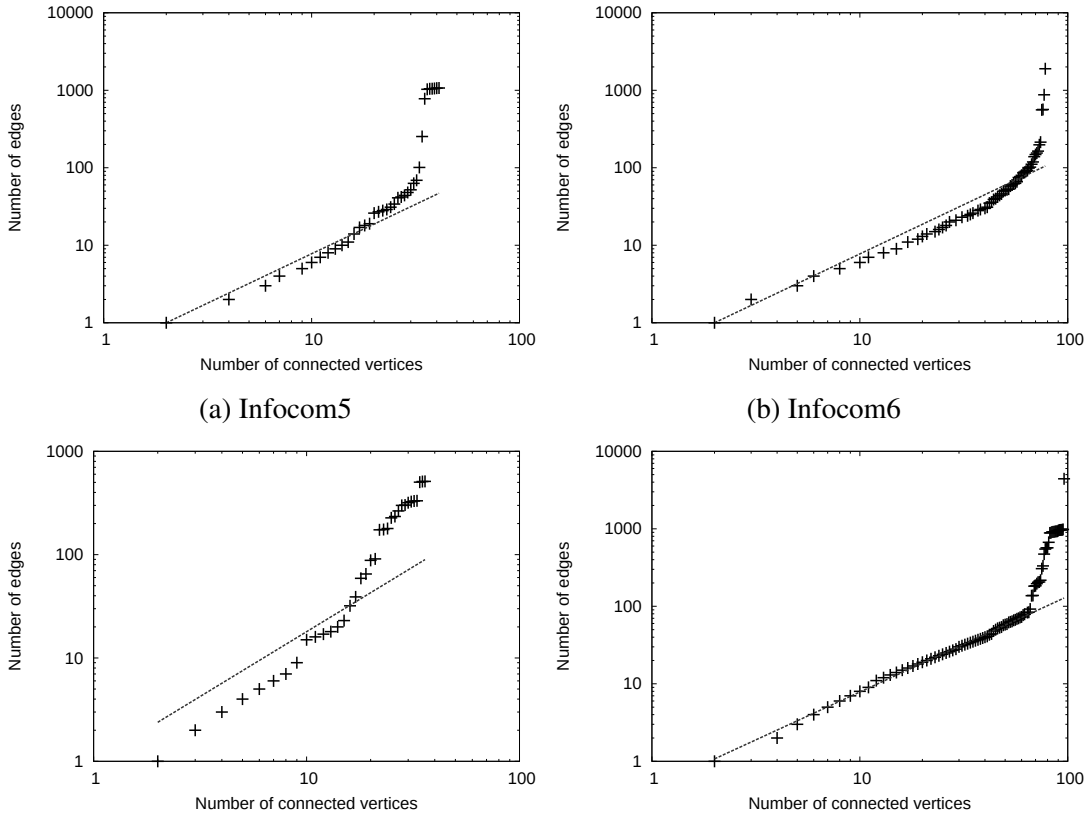


(a) Frame 4 (b) Frame 5 (c) Frame 6

Figure 5.3: The encounters present in the $4^{th}$, $5^{th}$, and $6^{th}$ hourly time frames of the Cambridge dataset that form strongly connected subgraphs within each time frame.

Now imagine that the Cambridge dataset has been split into a number of discrete, sequential time frames. For example, Figures 5.3a to 5.3c show some of the encounters that occur during the $4^{th}$, $5^{th}$, and $6^{th}$ hourly time frames of the Cambridge dataset (exactly which encounters will be discussed further in Section 5.5.3, as will the choice for time frame duration in Section 5.5.4). Detecting clusters within a short time is one example of spatio-temporal clustering as the clusters describe the encounters which occurred within a particular time frame.

This chapter will describe a spatio-temporal cluster detection algorithm which splits reality mining data into discrete time frames and cluster the data using the dynamic encounter graphs described in Section 2.5.3. Before introducing the algorithm in more detail, the next two sections will discuss the rate of change in reality mining datasets and previous work into spatio-temporal clustering.




(a) Infocom5 (b) Infocom6

(c) Cambridge (d) Reality

Figure 5.4: Each reality mining dataset tested has a DPL exponent of roughly 1.25 until 70% of the PMWDs are present in a monotonic encounter graph.

## 5.3 Human encounter patterns

This section will talk more about changing human encounter patterns and offer new temporal analysis of some reality mining datasets. The emphasis is on looking for temporal characteristics other than the bursty patterns already seen in Section 2.4 that are important to the design of new spatio-temporal cluster detection algorithms.

As monotonic encounter graphs generated from reality mining experiments grow, the rate at which new edges are attached to the graph follows a specific Densification Power Law (DPL) exponent [Leskovec et al., 2005] which is discussed in Section 2.6.2. Figure 5.4 shows all of the reality mining datasets tested densifying at a fairly consistent rate. The DPL exponent $\kappa$ is around 1.25 in all of the tested cases. In the Infocom datasets $\kappa$ is 1.27, and in the Cambridge and Reality datasets $\kappa$ is 1.25 and 1.24 respectively. What this tells us is that despite the number of encounters between reality mining participants being different in each dataset (See Table 1.1), the rate at which new PMWDs are encountered compared to known PMWDs is consistent across these datasets.

| | **Infocom5** | **Infocom6** | **Cambridge** | **Reality** |
|---|---|---|---|---|
| $1^{st}$ **quarter** | 0.3892 | 0.3549 | 0.0122 | 0.0003 |
| $2^{nd}$ **quarter** | 0.4049 | 0.0447 | 0.1754 | 0.0011 |
| $3^{rd}$ **quarter** | 0.0173 | 0.3116 | 0.0852 | 0.0019 |
| $4^{th}$ **quarter** | 0.4086 | 0.4683 | 0.0113 | 0.0012 |

Table 5.1: The probability of an encounter between participants in the $1^{st}$, $2^{nd}$, $3^{rd}$, and $4^{th}$ quarter of each day.

As well as the densification of reality mining datasets, there have been various other works which hint at the rate of change in human encounter patterns. Natarajan et al. [Natarajan et al., 2007] showed that the more time that has elapsed since the last encounter between two PMWDs, then the lesser the chance of an encounter between them in the future. This is further indication that the temporal relevance of aggregated monotonic clusters will diminish over time, and that spatio-temporal clusters used for opportunistic message forwarding in PSNs should have a limited lifespan.

The distribution of inter-encounter times in reality mining datasets has been shown to differ greatly between time frames that are 3 hours long [Chaintreau et al., 2005]. Table 5.1 expands on this by showing the changing probability of an encounter between any two PMWDs within 6 hour time frames. The different probability for each quarter day also confirms the observations made by Leung et al. [Leung et al., 2011] that network measurements change during different time periods, and it hints that cluster membership will be significantly different if calculated separately every 6 hours.

## 5.4 Existing spatio-temporal clustering algorithms

Now that some of the dynamic characteristics of human encounter patterns have been introduced, this section will describe work being done elsewhere on detecting spatio-temporal clusters in reality mining data.

Pietilainen and Diot detected spatial clusters within 60 second time frames and called these snapshot clusters. They also found a correlation between snapshot clusters that occur within several time frames, called temporal communities (see Figure 5.5), and relationships such as friendship and home city [Pietilainen and Diot, 2012]. Pietilainen and Diot then went on to show that social encounters between PMWDs that spend the most time within temporal communities do not impact on message delivery performance as much as non-social encounters. In fact, it has been shown on numerous other occasions that non-social encounters significantly outnumber social encounters in reality mining experiments [Lambiotte et al., 2008, Zyba et al., 2011, Allamanis et al., 2012, Gaito et al., 2012], which hints that temporal community detection will not help to improve opportunistic message delivery probability within PSNs.

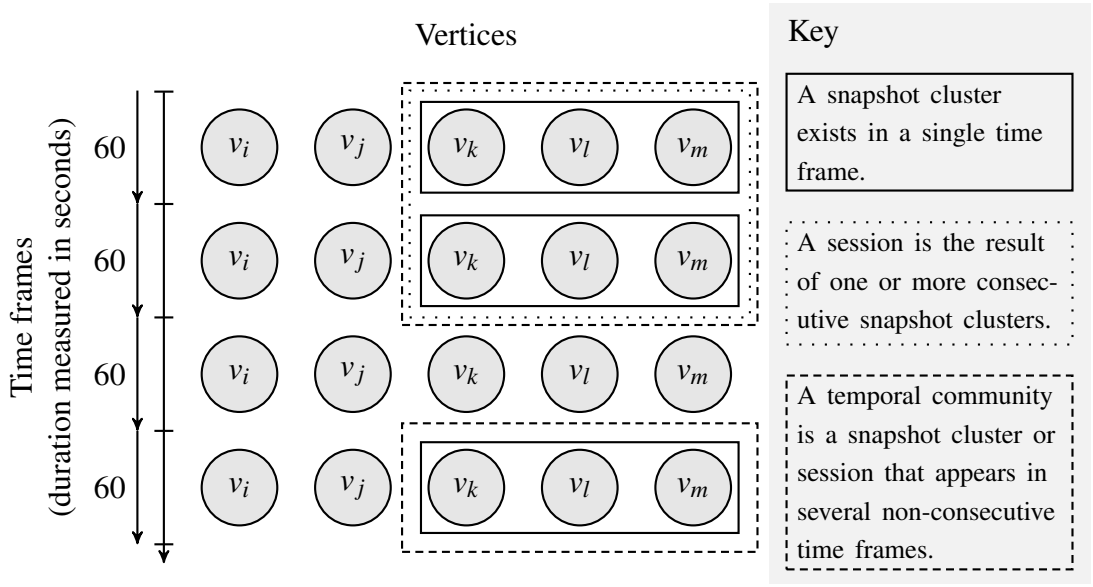



Figure 5.5: A series of 60 second time frames (from top to bottom). Each time frame contains the same five PMWDs. Also shown are examples of Pietilainen's spatio-temporal clusters [Pietilainen and Diot, 2012], which are calculated from the encounters between PMWDs that occur within each time frame (encounters are not shown).

Pietilainen and Diot also reported that 30-40% of sessions (snapshot clusters that span consecutive time frames, see Figure 5.5) lasted 10 minutes or more in the campus datasets, and that sessions in the Reality dataset are long lived. The median being 28 minutes with 25% lasting more than 1 hour [Pietilainen and Diot, 2012], possibly due to the campus timetable.

Natarajan et al. took a different approach to spatio-temporal cluster detection and looked for encounters between PMWDs which are longer than a familiar threshold, and which overlap to form "meetings" [Natarajan et al., 2007]. Figure 5.6 shows one example of a meeting from the point of view of one PMWD $v_i$. In this example, an encounter with $v_j$ overlaps with the encounter with $v_k$. The meeting time is therefore the total time from the start of the first encounter with $v_j$ to the end of the encounter

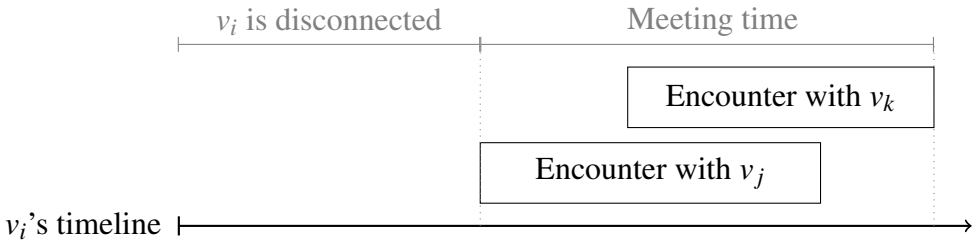



Figure 5.6: A single meeting for $v_i$ as defined by Natarajan et al.

with $v_k$. Using this definition Natarajan et al. found that mean meeting duration is 17 minutes in the Singapore reality mining dataset [Natarajan et al., 2007].

It is clear from Figures 5.5 and 5.6 that the methods proposed by Pietilainen and Natarajan will detect different patterns of human behaviour, and that the conclusions reached using the two methods are not directly comparable. Sessions as defined by Pietilainen and Diot are used to detect clusters that exist over multiple time frames, whilst meetings as defined by Natarajan et al. describe chains of encounters.

This chapter aims to uses spatio-temporal clusters for efficient message delivery in PSNs. In view of that, the next section will describe a centralised method for detecting spatio-temporal clusters that are formed when encounters create end-to-end paths within a group of PMWDs. The proposed approach will also allow for PMWDs moving between social groups to join new clusters without clusters monotonically increasing in size during the experiment.

## 5.5 Expectation-based spatio-temporal clustering

This section introduces two novel algorithms for spatio-temporal cluster detection in reality mining datasets. Both approaches analyse the static graphs formed within time frames of the dynamic encounter graphs that are described in Section 2.5.3, and both approaches are aimed at detecting spatio-temporal clusters as quickly as possible in the time series of encounter information from reality mining data:

1. Single frame Expectation-Based Spatio-temporal (SEBS) clusters are made up of connected vertices in static graphs formed during discrete time frames. The aim of SEBS cluster detection is to be able to detect when groups of nearby PMWDs experience a sudden rise in encounter duration with each other (such as at a concert or during a long journey on a crowded train).

2. The second spatio-temporal clustering algorithm being offered in this chapter is for the detection of Multiple frame Expectation-Based Spatio-temporal (MEBS) clusters. MEBS clusters are formed when edges between a group of vertices form a strongly connected subgraph (strongly connected subgraphs will be introduced more formally in Section 5.5.3), and when the strongly connected subgraph is present across multiple consecutive time frames. Thus MEBS clusters are similar to the sessions described by Pietilainen and Diot and which are illustrated in Figure 5.5.

The SEBS and MEBS clustering algorithms belong to the *expectation-based* category of spatio-temporal clustering algorithms [Neill, 2006, Orlinski and Filer, 2012a]. These clustering algorithms detect spatio-temporal clusters from patterns of behaviour that are out of the ordinary compared to the recent past. Critical to the operation of expectation-based spatio-temporal clustering algorithms are the metrics and methods used in baseline calculation, which are described in the following two subsections.

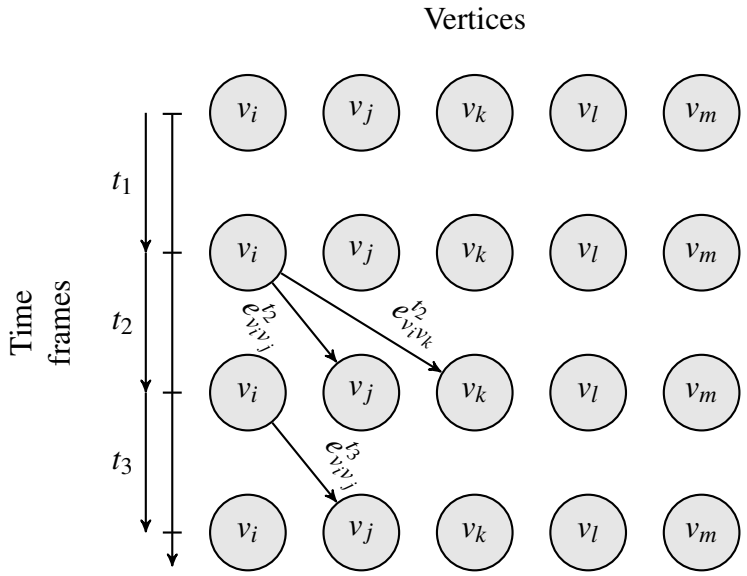



Figure 5.7: A dynamic encounter graph between time frames $t_1$ and $t_3$.

### 5.5.1 Edges weights and metrics

Due to the frequent disruption to encounters experienced because of the parking lot problem, it is more difficult to meaningfully count new encounters than it is to estimate cumulative encounter duration. Therefore, the weight of edges between vertices in SEBS and MEBS cluster detection represents the cumulative encounter duration within each discrete time frame. For example, the cumulative encounter duration for a vertex $v_i$ with vertex $v_j$ during time frame $t_2$ as shown in Figure 5.7 is $e^{t_2}_{v_iv_j}$. The mean cumulative encounter duration during $t_2$ for vertex $v_i$ in this case is shown in Equation 5.1. The mean cumulative encounter duration for a vertex is also referred to as the metric $m$ in this thesis, and $m$ will be used when calculating baselines in the next section.

$$m_i^{t_2} = \frac{e^{t_2}_{v_iv_j} + e^{t_2}_{v_iv_k}}{2} \tag{5.1}$$

### 5.5.2 Calculating vertex baselines

SEBS and MEBS cluster detection involves calculating expected values for metrics called *baselines* from the time series of previous values. This removes the need to manually assign thresholds when choosing vertices to cluster together.

Baselines for the current time frame are calculated at the end of the previous time frame. To make describing this process easier, each time frame of length $l$ is labelled using the time series $t_1, t_2, \ldots, t_{(n-1)}, t_n$, where $t_1$ is the first time frame, $t_{(n-1)}$ is the last complete frame, and $t_n$ is the current frame. This allows the baseline for a vertex $v_i$ for the current frame to be labelled $b_i^{t_n}$.

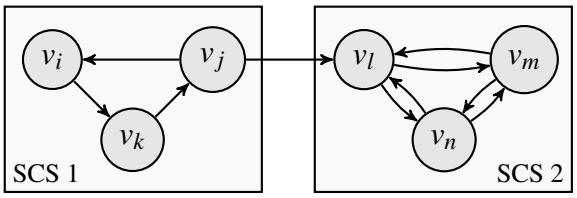



Figure 5.8: A directed static graph with two Strongly Connected Subgraphs (SCS). Note that within each SCS a path exists between each vertex.

In SEBS and MEBS cluster detection, the baseline calculation for a vertex $v_i$ for the current time frame $t_n$ is done by calculating the mean of the $m$ values from the past $w$ complete time frames as in Equation 5.2.

$$b_i^{t_n} = \frac{m_i^{t_{(n-1)}} + m_i^{t_{(n-2)}} + \cdots + m_i^{t_{(n-w)}}}{w} \tag{5.2}$$

As the number of encounters may vary within the last $w$ time frames, taking the mean of the mean in this way produces an estimated mean. Calculating the estimated mean rather than the true mean may seem a weakness at first, but the baseline is just a guide. This calculation is preferable to setting manual familiar thresholds for each vertex in each experiment, and does not require that a record be kept of all edge weights over a long period of time.

The size of $w$ should be chosen carefully as this will alter the number of time frames from which the baseline is calculated, and thus the baseline itself. Long term trends upwards or downwards in the cumulative encounter durations are countered by choosing values for $w$ which are greater than 1, but smaller than the total length of the experiment $L$ divided by the time frame length $l$, $1 < w < (L/l)$. For example, in order to detect the bursts of cumulative encounter duration in reality mining datasets seen in Section 2.5.3, the result of $w$ multiplied by $l$ should equal no more than 12 hours. Otherwise baselines will be calculated over a period which spans multiple bursts.

### 5.5.3 Strongly connected subgraphs

As encounters between PMWDs are directional [Madan et al., 2010], the static graphs used to represent them can contain directed edges. Directed static graphs can also contain *strongly connected subgraphs* [Cormen et al., 2011], which are sections of a static graph where there is a path from each vertex to every other vertex. Two example strongly connected subgraphs are shown in Figure 5.8.

Figure 5.3 shows the encounters between PMWDs that form strongly connected subgraphs in the $4^{th}$, $5^{th}$, and $6^{th}$ hourly time frames of the Cambridge dataset. In this chapter the hypothesis that strongly connected subgraphs generated from past encounters act as an indication of strong bonds between PMWDs that will persist into the near future is being explored. Figure 5.3 can also be used to illustrate this point, vertex 11 and 33 are reachable by the different strongly connected subgraphs in each of these 3 hourly time frames. Thus, any message given to vertex 11 for vertex 33 in the $4^{th}$

hour will be delivered by the $5^{th}$ hour. Furthermore, Figure 5.3 also shows that messages given to either vertex 11 or 33 to be delivered to vertex 27 in the $4^{th}$ hour will be delivered in the $6^{th}$ hour.

### 5.5.4 Length of time frames

Calculating the length of time frames in SEBS and MEBS clustering is application specific, and it is not feasible to exhaustively compare all possible resolutions in order to satisfy some criteria. However, it is possible to determine suitable time frame lengths using the time needed to form strongly connected subgraphs as a guide.

In order to form strongly connected subgraphs within time frames, the SEBS and MEBS clustering algorithms use longer time frames than the 60 second frames used by Pietilainen and Diot [Pietilainen and Diot, 2012]. This is so that the algorithms can aggregate enough encounters to form large static graphs by the end of each time frame. Figure 5.9 tells us that there is a 0.95 probability that strongly connected subgraphs take less than 1 hour to form in both of the Infocom datasets, whereas in Reality the probability of a strongly connected subgraph forming in less than an hour is just 0.68.

The mechanism used to compile Figure 5.9 is unusual, and therefore requires some introduction. To generate the data used in Figure 5.9, encounters within a dataset are aggregated into a non-monotonic encounter graph in chronological order. If a new edge results in a new strongly connected subgraph, then the time the subgraph took to form is calculated as being the time from when the first edge is added to the strongly connected subgraph up to the current time. Furthermore, in order that formation time in Figure 5.9 relates only to new strongly connected subgraphs, all of the edges in the newly detected strongly connected subgraph are removed from the encounter graph once the formation time has been calculated and stored.

Figure 5.9 hints that SEBS and MEBS clustering requires a minimum time frame length of 1 hour. This is sufficient to discover 80% of the strongly connected subgraphs formed in the Cambridge dataset (using the method described in the previous paragraph), but only 68% of the strongly connected subgraphs in the Reality dataset. Therefore, time frames of 6 and 12 hours will also be used in the analysis in Section 5.6. In the Reality dataset only 85% of strongly connected subgraphs form within 12 hour time frames, but it is difficult to extend time frame length over 12 hours in SEBS and MEBS cluster detection because of the baseline calculation. If time frames longer than 12 hours are used, then baselines would be calculated over a period of time which overlaps with two bursts of activity.

### 5.5.5 Score function

SEBS and MEBS clustering also involves calculation of the *significance* of strongly connected subgraphs. Significance is determined by collectively comparing the metrics and baselines of vertices within strongly connected subgraphs. This is done so the SEBS and MEBS clustering algorithms are capable of treating smaller strongly connected subgraphs that are part of larger ones as separate clusters, so long as metrics are

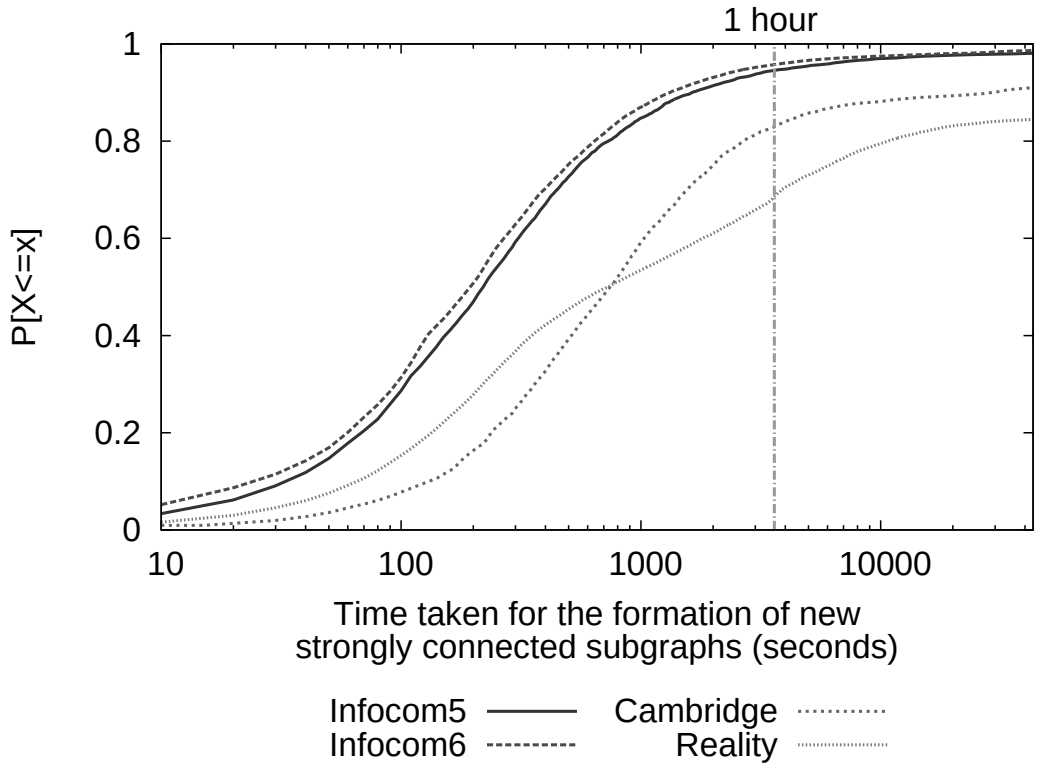



Figure 5.9: Cumulative probability distribution of the time taken to form new strongly connected subgraphs in the tested reality mining datasets.

significantly higher than their respective baselines.

The process of assessing the significance of strongly connected subgraphs in SEBS and MEBS clustering involves Neill's score function [Neill, 2006] shown in Equation 5.3, where $s_x^{t_n}$ is one strongly connected subgraph at time frame $t_n$. Generally speaking, Neill's score function can be used to compare the metrics within a spatial region against its baselines. Less generally, Neill's score function is being used here to assess the significance of strongly connected subgraphs within static graphs.

$$F_p(s_x^{t_n}) = (\frac{M}{B})^M e^{B-M} \text{ if } M > B, \text{ and } F_p(s_x^{t_n}) = 1 \text{ otherwise.} \tag{5.3}$$

The baseline for a strongly connected subgraph (B) is calculated by summing the baselines from the individual vertices contained within the subgraph. The baseline for the strongly connected subgraph $s_x^{t_n}$ at time frame $t_n$ is then $B = \sum_{v \in s_x^{t_n}} b^{t_n}$, and the metric for $s_x^{t_n}$ is $M = \sum_{v \in s_x^{t_n}} m^{t_n}$. The mathematical constant $e$ is present in the score function following the simplification from the score function's original form as discussed on pages 36 and 37 of [Neill, 2006].

The score function as shown in Equation 5.3 is identical to the score function detailed in [Neill, 2006], where it is used to assess the significance of spatial clusters in the detection of disease epidemics. The next two subsections will describe how the

score function is being used in this thesis in order to detect if strongly connected subgraphs contained within other strongly connected subgraphs are more significant than their parent or parents, and warrant inclusion in the analysis in Section 5.6.

### 5.5.6 SEBS clustering

In this chapter, SEBS cluster detection is performed on the dynamic encounter graphs generated from reality mining datasets. Whilst building dynamic encounter graphs from the datasets, SEBS cluster detection creates clusters from the static graphs produced at the end of each time frame.

For example, at the end of time frame $t_n$ the algorithm will attempt to detect strongly connected subgraphs with 3 or more vertices from the newly formed static graph. Then the vertices of each strongly connected subgraph are tested to see if their metric values are greater than their baselines, i.e. the significance of all the vertices within a detected strongly connected subgraph $s_x^{t_n}$ are tested using the condition $\forall v_i^{t_n} \in s_x^{t_n} : m_i^{t_n} > b_i^{t_n}$, where $|s_x^{t_n}| \geq 3$.

The score function $F_p(s_x^{t_n})$ is then used to remove less significant strongly connected subgraphs from the analysis with the check, $\forall S^{t_n} \supset s_x^{t_n} : F_p(s_x^{t_n}) \geq F_p(S^{t_n})$. This check ensures that all strongly connected subgraphs detected are not structurally weaker than their parent strongly connected subgraphs ($S^{t_n}$) in terms of the metric used.

Once less significant strongly connected subgraphs have been removed from the analysis using the score function, those remaining are referred to as SEBS clusters. To summarise, in order to classify a strongly connected subgraph $s_x^{t_n}$ which occurred in the time frame $t_n$ as a SEBS cluster, $s_x^{t_n}$ must satisfy the conditions $\forall v_i \in s_x^{t_n} : m_i^{t_n} > b_i^{t_n}$, $|s_x^{t_n}| \geq 3$, and $\forall S^{t_n} \supset s_x^{t_n} : F_p(s_x^{t_n}) \geq F_p(S^{t_n})$. Finally, new baselines are calculated for each vertex at the end of the SEBS detection process.

### 5.5.7 MEBS clustering

Strongly connected subgraphs may only need to persists over multiple consecutive time frames in order to be labelled a MEBS cluster. A simple way of summarising the MEBS clustering process is that it searches $f$ consecutive time frames where $f > 1$, and looks for strongly connected subgraphs of at least 3 vertices which are present in each time frame.

As well as requiring that strongly connected subgraphs are at least 3 vertices in size, MEBS cluster detection also requires that the strongly connected subgraphs satisfy the following condition in order to discard less significant clusters for the analysis in Section 5.6.2. In order to be classified as a MEBS cluster, a strongly connected subgraph which spans the interval $t_{(n-(f-1))} \ldots t_n$ (called $s_x$), must have a higher score in at least one frame than every strongly connected subgraph which is a super-set of $s_x$ that also spans the interval $t_{(n-(f-1))} \ldots t_n$. In other words, for a strongly connected subgraph which spans the interval $t_{(n-(f-1))} \ldots t_n$ to be considered a MEBS cluster, there

must exist a time frame called $t_{max}$ in the interval $t_{(n-(f-1))} \dots t_n$ where $\forall S^{t_{(n-(f-1))} \dots t_n} \supset s_x^{t_{(n-(f-1))} \dots t_n} : F_p(s_x^{t_{max}}) \geq F_p(S^{t_{max}})$. It should also be added that if there is no super-set of $s_x$ spanning the interval $t_{(n-(f-1))} \dots t_n$, then $s_x$ will also be considered a MEBS cluster.

## 5.6 Expectation-based spatio-temporal cluster analysis

Pietilainen and Diot showed that temporal communities which exist in reality mining datasets often have less than 10 members [Pietilainen and Diot, 2012]. This section will present analysis on the timing and size of SEBS and MEBS clusters in order to offer further insights into the way people congregate in the reality mining datasets. It is important to mention that the Reality dataset as presented in this analysis is truncated, and only the data between the time-stamps 1094545041 and 1111526856 is used. This is because there is no significant activity before and after these times respectively.

### 5.6.1 Analysis of SEBS clusters

SEBS cluster analysis has been conducted using frame lengths $l$ of 1,6, and 12 hours. Baseline calculation is changed between experiments to see its effects on clustering, and observations are also made for different $w$ values of 2, 4, 12, 18, and 24.

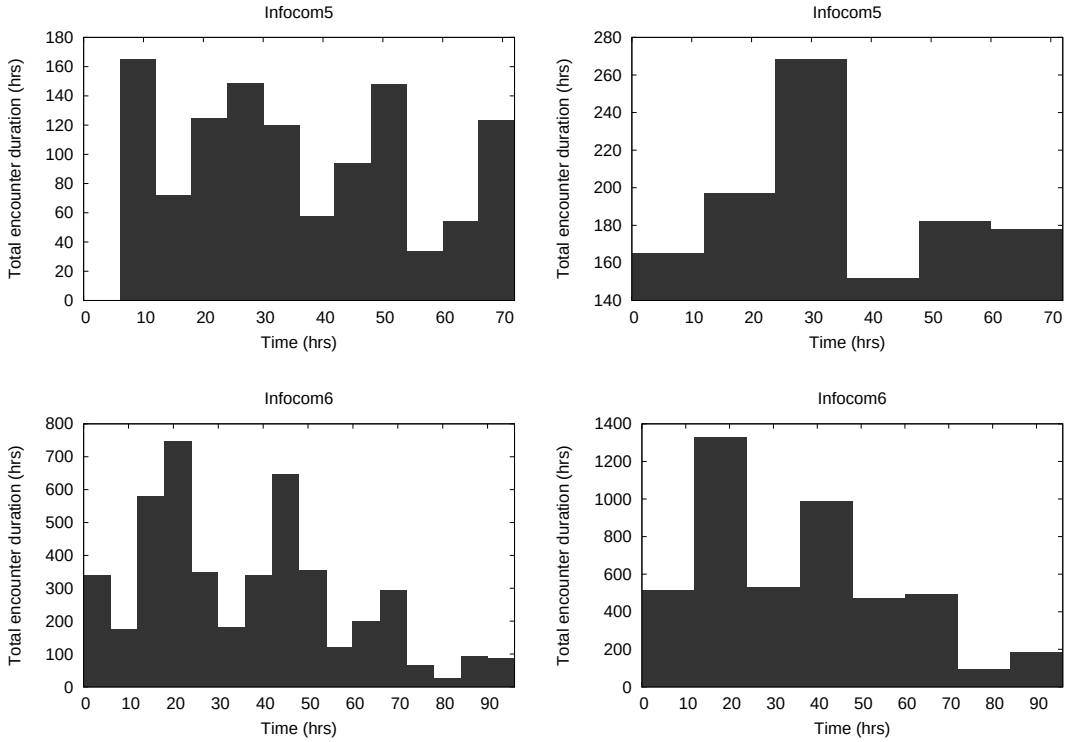
#### SEBS cluster detection times

Table 5.2 shows the percentage of time frames where new SEBS clusters are detected in the datasets tested. It is important to remember that the percentage of time frames containing SEBS clusters may be lower than the number of time frames where strongly connected subgraphs are present because of the baseline comparison. Likewise, time frames where cumulative encounter duration is greater than the previous frame may not contain SEBS clusters because no strongly connected subgraphs are formed.

As time frame length is increased, there are fewer significant changes in total encounter duration between frames. This is shown visually in Figure 5.10 where different levels of data aggregation are shown side by side. Despite over 95% of strongly connected subgraphs forming within 3600 seconds in the Infocom5 dataset, fewer bursts of cumulative encounter duration mean that there are fewer opportunities to form SEBS clusters when time frames are 43200 seconds long. Hence why Table 5.2 shows the percentage of frames containing SEBS clusters falling to 0% in the Infocom5 experiments.

#### SEBS cluster size

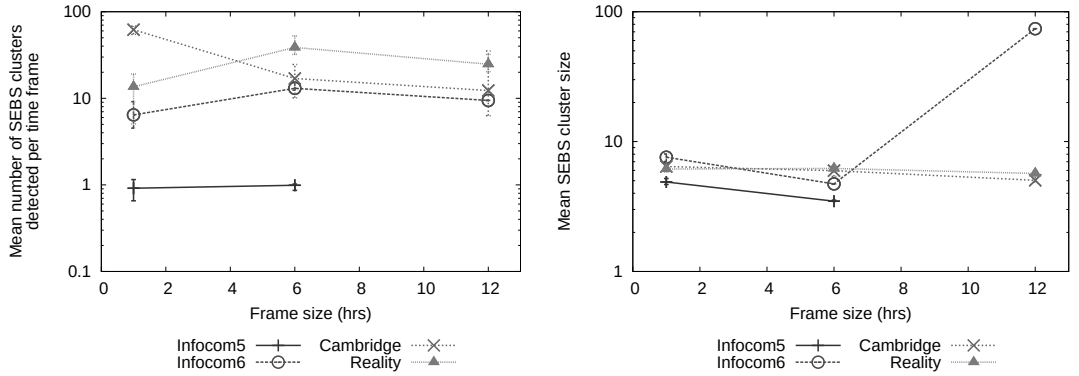
When Infocom6 is excluded from the mean cluster size calculation, the mean SEBS cluster size detected for all datasets is around 5 PMWDs with no consistent change seen when increasing time frame length. However, SEBS cluster size in the Infocom6 experiment can be seen to increase sharply in Figure 5.11b. This is because a small




(a) 6 hour time frames.

(b) 12 hours time frames.

Figure 5.10: The number of time frames where metrics are higher than the previous time frames decreases as time frame length increases.




(a) Number of SEBS clusters

(b) Mean size of SEBS clusters

Figure 5.11: Mean size and number of the SEBS clusters detected in each of the reality mining dataset tested.

| | **Infocom5** | **Infocom6** | **Cambridge** | **Reality** |
|---|---|---|---|---|
| **l = 3600 seconds, w = 2** | 19.737 | 14.737 | 10.219 | 19.504 |
| **l = 3600 seconds, w = 4** | 21.053 | 17.895 | 13.139 | 21.030 |
| **l = 3600 seconds, w = 12** | 23.684 | 25.263 | 18.248 | 24.422 |
| **l = 3600 seconds, w = 18** | 30.263 | 28.421 | 18.248 | 24.698 |
| **l = 3600 seconds, w = 24** | 30.263 | 26.316 | 17.153 | 24.083 |
| **l = 21600 seconds, w = 2** | 25.000 | 20.000 | 22.222 | 37.023 |
| **l = 21600 seconds, w = 4** | 33.333 | 20.000 | 20.000 | 32.824 |
| **l = 21600 seconds, w = 12** | 33.333 | 20.000 | 15.556 | 32.316 |
| **l = 21600 seconds, w = 18** | 33.333 | 20.000 | 15.556 | 32.952 |
| **l = 21600 seconds, w = 24** | 33.333 | 20.000 | 17.778 | 32.570 |
| **l = 43200 seconds, w = 2** | 0.000 | 14.286 | 9.091 | 34.351 |
| **l = 43200 seconds, w = 4** | 0.000 | 14.286 | 13.636 | 34.097 |
| **l = 43200 seconds, w = 12** | 0.000 | 14.286 | 18.182 | 31.298 |
| **l = 43200 seconds, w = 18** | 0.000 | 14.286 | 13.636 | 33.588 |
| **l = 43200 seconds, w = 24** | 0.000 | 14.286 | 13.636 | 33.333 |

Table 5.2: Percentage of time frames where new SEBS cluster are detected in the different datasets and using different values for $l$ and $w$.

number of very large SEBS clusters are created early on in the Infocom6 experiment when $l = 43200$ seconds, after which time participation in the experiment appears to diminish (see Figure 2.3). This is known as the *premature clustering problem*, and can be caused in SEBS clustering by a combination of effects including:

1. Early densification, and a low mixing rate in the latter stages of an experiment.
2. High metric values at the start of an experiment, and ever decreasing metric values thereafter. This is evident from the number of encounters per hour as shown back in Figure 2.3a which shows there tends to be fewer encounters as the Infocom6 experiment progresses.
3. Time frames that encompass one or more complete burst cycles, resulting in similar baselines and metric values.

### 5.6.2 Analysis of MEBS clusters

Multiple frame Expectation-Based Spatio-temporal (MEBS) cluster detection can be used to tell us about the duration of clusters created from strongly connected subgraphs in reality mining data. The timing, size, and shape of MEBS clusters are very different

| | **Infocom5** | **Infocom6** | **Cambridge** | **Reality** |
|---|---|---|---|---|
| **l = 3600 seconds, f = 2** | 27 | 212 | 3012 | 15453 |
| **l = 3600 seconds, f = 3** | 10 | 116 | 692 | 2429 |
| **l = 3600 seconds, f = 4** | 2 | 52 | 45 | 290 |
| **l = 21600 seconds, f = 2** | 2 | 0 | 1 | 3024 |
| **l = 21600 seconds, f = 3** | 0 | 0 | 0 | 8 |
| **l = 21600 seconds, f = 4** | 0 | 0 | 0 | 1 |
| **l = 43200 seconds, f = 2** | 1 | 0 | 0 | 294 |
| **l = 43200 seconds, f = 3** | 0 | 0 | 0 | 9 |
| **l = 43200 seconds, f = 4** | 0 | 0 | 0 | 0 |

Table 5.3: There are different numbers of MEBS clusters in each dataset when using different variable values.

to that of SEBS because MEBS clustering detects strongly connected subgraphs that exist in sequential time frames.

The first three data columns of Table 5.3 can be used to compare the number of MEBS clusters detected over 2 hourly frames ($l = 3600, f = 2$) with those which span 3 or 4 time frames. It shows that MEBS clusters (which represent transient groups of humans) which last for longer than 3 hours are rare in all of the datasets tested.

1. In the Reality dataset only 15.7% of the MEBS clusters which span 2 hourly time frames exist for a third hour.

2. Less than 23% of MEBS clusters exist for more than 2 hourly time frames in the Cambridge dataset.

3. MEBS clusters in the experiments conducted at conferences tend to last longer than in the campus experiments. 37% of MEBS clusters exist for longer than 2 hours in the Infocom5 dataset, and 54% of MEBS clusters last for longer than 2 hours in the Infocom6 dataset.

There is also a separation between the data collected at conferences and on campus when looking at the size of MEBS clusters. Figure 5.12 shows that MEBS clusters tend to be larger in the campus scenarios than they do in reality mining experiments performed at conferences. The MEBS clusters in the Reality and Cambridge datasets are most likely to be around 6 PMWDs in size. Whilst in the Infocom5 dataset very few of the MEBS clusters detected are larger than 3 PMWDs, and only 50% are larger than 3 PMWDs in the Infocom6 dataset. This means that even though the Infocom6

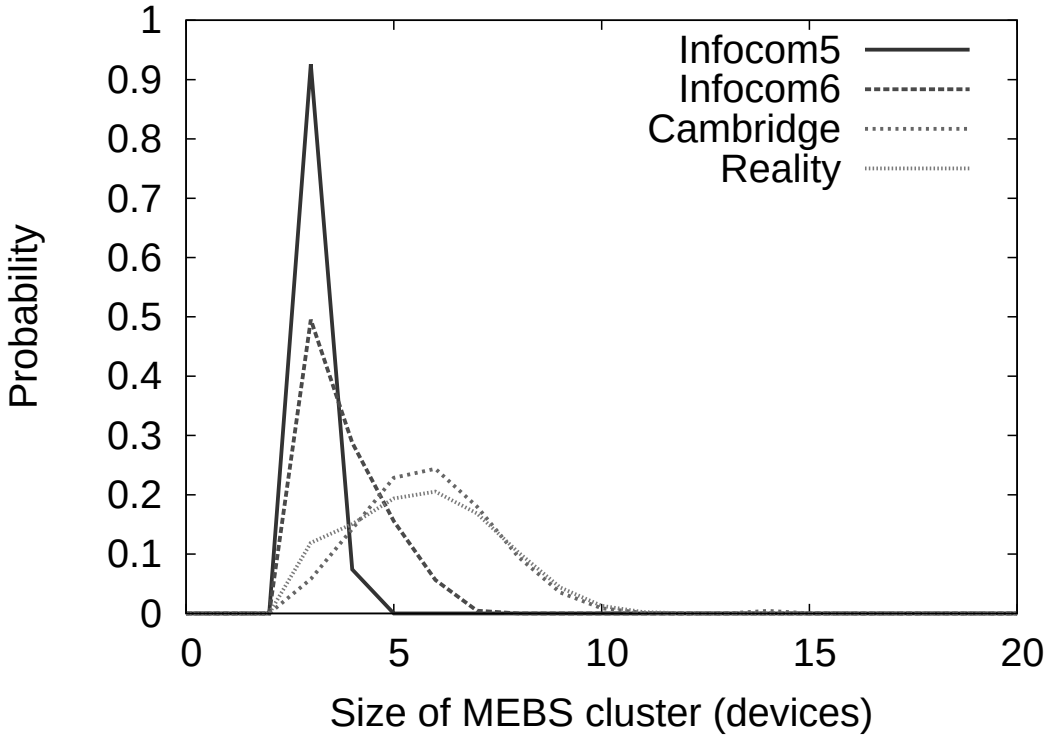



Figure 5.12: Probability that a MEBS cluster is a particular size in the tested datasets when $l$=3600 seconds and $f$=2.

dataset exhibits the premature clustering problem with SEBS cluster detection, recurring strongly connected subgraphs in Infocom6 are between 2-7 PMWDs in size and last no more than 2 to 3 hours.

By considering the duration and size of MEBS clusters, a picture emerges of small yet long lasting MEBS clusters in the conference datasets, with larger MEBS clusters which last for shorter periods in the campus wide experiments. One reason for the observations made about encounters at conferences is that small groups tend travel and stay together during conferences. This behaviour was also observed in [Hui and Crowcroft, 2008] and the groups were given the name of *affiliation communities*. The larger MEBS clusters seen in the campus scenarios are also easily explained; people tend to work individually on a university campus, apart from when they form large short lived clusters in places such as cafés, dinning halls, and lecture theatres.

## 5.7 Message delivery using SEBS clusters

This section explores the use of SEBS clusters as boundaries for message duplication in PSNs. MEBS clusters will not be tested for their message delivery properties here because they take longer to detect than SEBS clusters (at least two complete time frames). Thus any advantage of using MEBS clusters may have passed by the time

they are detected. However, future work may wish to explore wherever it is possible to detect recurring MEBS clusters, as this will enable opportunistic message delivery protocols to predict paths between PMWDs using recurring strongly connected subgraphs.

The proposed opportunistic message delivery algorithm based on SEBS clusters is compared to the non-clustering technique PRoPHETv2 [Grasic et al., 2011], and the distributed clustering and message delivery protocols Bubble [Hui et al., 2007a] and Nomads in order to gauge the benefits of using expectation-based spatio-temporal over aggregated monotonic clustering.

### 5.7.1 Experimental environment

The message delivery experiments in this chapter are similar to those in Section 4.2.1, but a different selection of reality mining datasets are used. This is because the location sensors that were used to infer encounters in the UCL1 experiment may be inaccurate indoors due to a loss of GPS signal, and each encounter in UCL1 was estimated to be 10 seconds long.

In the same way as is described in Section 4.2.1, the results in this chapter are provided by five random message generation patterns for each dataset/algorithm configuration. Message TTL is 1 hour in the Infocom5, Infocom6, and Cambridge datasets, and 1 day in the Reality dataset because of the sparsity of encounters. It is also important to note that only the first 35 days after the 1094545041 time-stamp are used in the message delivery experiments involving Reality. This is because the time needed to perform all of the necessary experiments over such a large time span would have been too long when using the full 246 days available, or the 197 days as used for cluster analysis in Section 5.6.1.

### 5.7.2 Message delivery mechanism for SEBS clusters

In the opportunistic message delivery experiments described in this chapter, PMWDs have access to SEBS clusters which are being generated by a centralised process. The PMWDs can access this information at anytime and from anywhere, but PMWDs will not be able to use SEBS clusters that are generated in the future.

So that the message delivery results of this chapter are comparable with those in Chapter 4, the message forwarding algorithm to be used with SEBS clusters is almost the same as that which is described in Section 4.4.3. As a result, the message forwarding algorithm will not attempt to look for end-to-end paths between PMWDs using overlapping SEBS clusters (which it could do because of the centralisation). Furthermore, as spatio-temporal clusters detected in short time frames are being used instead of aggregated monotonic clusters, the forwarding algorithm described in Algorithm 5 must decide which (if any) of a PMWD's previous spatio-temporal clusters are to be used in forwarding decisions.

**Algorithm 5** Message delivery algorithm used with SEBS clusters.

1. If the PMWD $v_i$ has a message to send to $v_k$ at time $t^x$, and $v_i$ encounters $v_j$ also at time $t^x$, then $v_i$ will check the most recent complete time frame before $t^x$ which has a SEBS cluster containing $v_j$. If such a time frame exists (called $t^x_{v_j}$), $v_i$ then checks to see if any SEBS cluster in $t^x_{v_j}$ contains both $v_j$ and $v_k$:
   (a) If any SEBS cluster in $t^x_{v_j}$ contains both $v_j$ and $v_k$, copy the message to $v_j$.
   (b) Otherwise do nothing.

### 5.7.3 Message delivery performance

The message delivery results using SEBS clusters shown in Figure 5.13 show that delivery probability does not change significantly with different $l$ and $w$ values. The exception being the Infocom6 dataset where message delivery probability increases sharply when frame sizes of 12 hours are used. In this case, the premature cluster problem leads to almost all of the PMWDs being grouped into a single SEBS cluster at the end of the second time frame, and no more clusters are produced after.

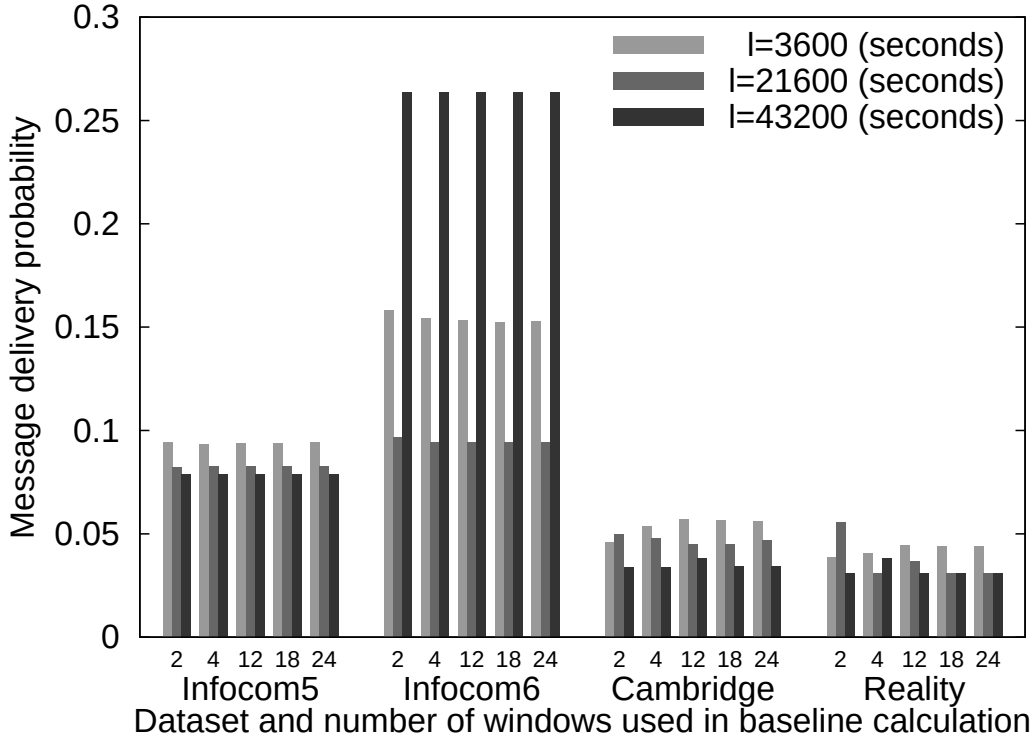



Figure 5.13: Message delivery probability using SEBS clusters. Detected using frame lengths $l$ of 3600, 21600, and 43200 seconds, and baseline calculation ($w$) over 2, 4, 12, 18, and 24 frames.

| | Message delivery probability | Overheads |
|---|---|---|
| **Bubble K-Clique** | 0.114 | 20.680 |
| **Bubble Simple** | 0.115 | 20.764 |
| **SEBS (l = 3600 seconds)** | 0.086 | 10.738 |
| **SEBS (l = 21600 seconds)** | 0.065 | 5.125 |
| **SEBS (l = 43200 seconds)** | 0.102 | 18.876 |
| **PRoPHETv2** | 0.147 | 25.965 |
| **Nomads** | 0.172 | 56.389 |

Table 5.4: Mean delivery probability and overheads for the different protocols.

Figure 5.14 shows the number of messages delivered to their final destinations and efficiency over time for opportunistic message delivery using SEBS clusters, Nomads, Bubble, and PRoPHETv2. The results for SEBS clustering are taken from the experiments which give the greatest message delivery probability from Figure 5.13. The other protocols have been afforded the same benefit, and the results presented in Figure 5.14 are of their best possible message delivery probability in these tests.

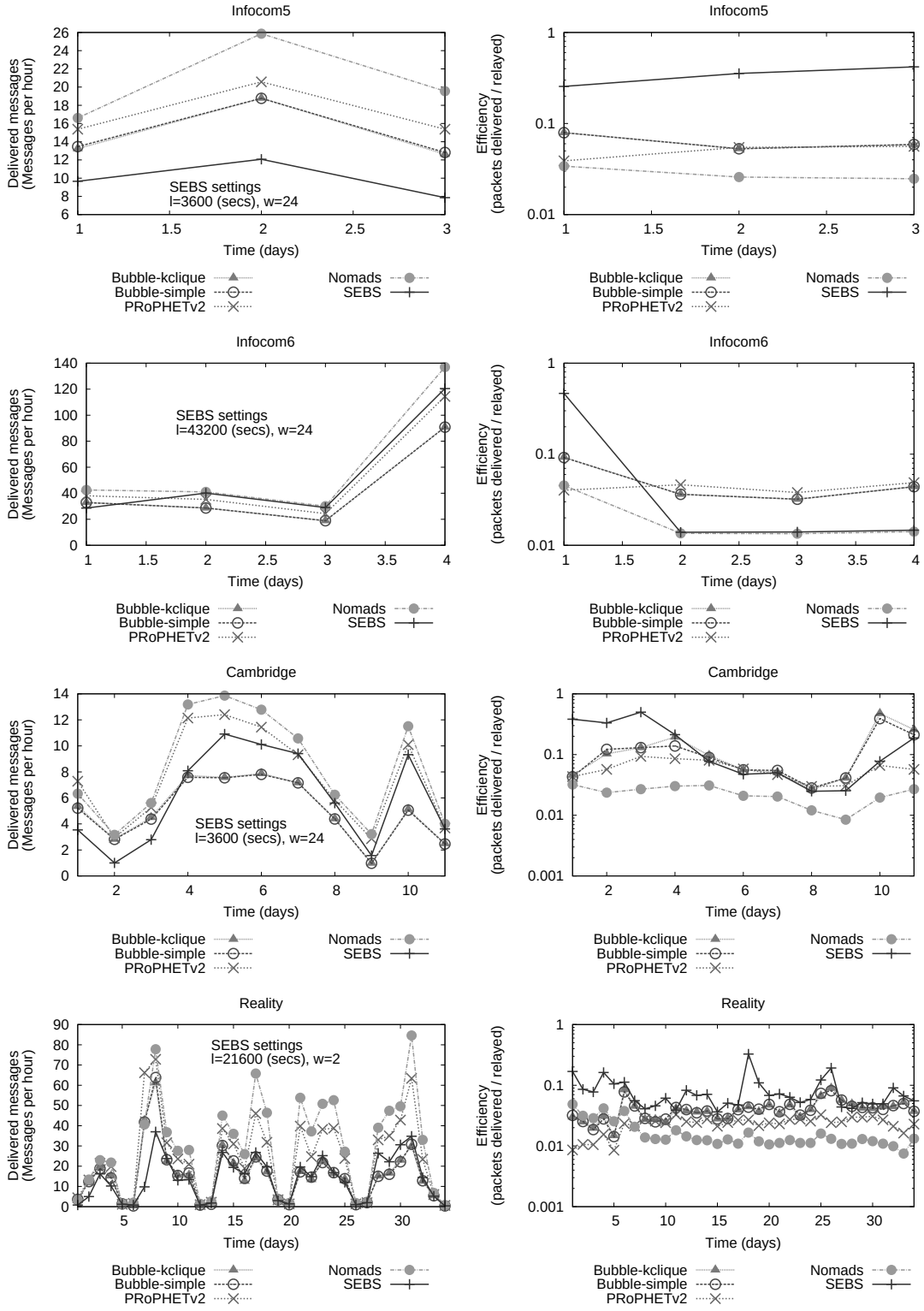
Generally, the results in Figure 5.14b show that the message delivery performance of SEBS cluster based forwarding is more efficient compared with the other protocols, but there are some important exceptions. The large clusters formed at the start of the Infocom6 experiment when $l = 43200$ seconds causes many duplicate packets and low efficiency. In contrast, message delivery probability for SEBS clusters in the Infocom5 dataset is very low because of the small size of SEBS clusters (around 4 PMWDs).

With the help of Figure 5.14a it can also be seen that the number of messages delivered in the latter stages of the Cambridge and Reality experiments is higher when using SEBS clusters than when using Bubble. This is because SEBS clustering tries to detect new spatio-temporal clusters throughout an experiment, whereas Bubble appears to show diminishing ability to react to changing human encounter patterns as an experiment progresses because of its monotonic clustering.

Table 5.4 combines the message delivery probability and overheads for the different experiments described in Section 5.7.1 and presents the mean of each value. The picture that emerges in Table 5.4 is one where opportunistic message delivery using SEBS clusters has the lowest message delivery probability, and Nomads has the highest. This is partly due to the small size of SEBS clusters as seen in Figure 5.11b where the mean SEBS cluster size is 5 PMWDs.

## 5.8 Summary

This chapter described two novel algorithms for detecting expectation-based spatio-temporal clusters within dynamic encounter graphs. The SEBS and MEBS clustering algorithms make practical use of strongly connected subgraph detection within discrete




(a) Delivery (b) Efficiency

Figure 5.14: Mean number of messages delivered per hour and efficiency over time for the SEBS cluster delivery protocol and others.

time frames, and baseline calculations with which encounter metrics can be tested for significance. Analysis on the resulting SEBS clusters also highlighted the premature clustering problem which future expectation-based spatio-temporal clustering algorithms should attempt to avoid.

The long time frames of 1, 6, and 12 hours used in SEBS cluster detection can exacerbate the premature cluster problem by not giving the algorithm enough time to alter baselines to represent a seasonal mean. However, long time frames are needed in SEBS cluster detection in order to aggregate enough encounters together to form strongly connected subgraphs (see Section 5.5.3). Future iterations of the SEBS clustering algorithm may involve separating the baseline calculation and strongly connected subgraph generation into two separate threads which have their own time frame lengths.

Also in this chapter, SEBS clusters are used in messages forwarding decisions in a prototype message delivery protocol. As expected overheads caused by duplicate messages are shown to be lower than in protocols which use aggregated monotonic clusters in forwarding decisions. However, attempting to assert that spatio-temporal clusters can be used to deliver messages in a PSN would be premature at this stage. SEBS cluster based message delivery had a tendency to exhibit smaller delivery probabilities than the other protocols tested. This suggests that the requirement that SEBS clusters be made from strongly connected subgraphs was not a wise choice as this limited cluster size. Therefore, a mechanism which creates larger expectation-based spatio-temporal clusters is explored in the next chapter.

PMWDs which make up the PSNs explored in this thesis have limited transmission range, and cannot rely on a centralised data store for clustering information as is done in Section 5.7. Therefore, a further challenge which is tackled in the next chapter is the distribution of expectation-based spatio-temporal cluster detection algorithms in order that they may be used in PSNs and other distributed systems.

# Chapter 6

# Distributed expectation-based spatio-temporal cluster detection

The previous chapter proposed two new centralised algorithms with which to detect expectation-based spatio-temporal clusters in dynamic encounter graphs. This chapter details work toward a distributed algorithm which can be used by PMWDs to detect expectation-based spatio-temporal clusters in a PSN. In addition, three new multiple copy opportunistic message delivery protocols which use spatio-temporal clusters in forwarding decisions are proposed.

The clustering algorithm outlined in this chapter will detect clusters similar to SEBS clusters. However, the strongly connected subgraph constraint has been dropped in order to produce larger clusters that are more suited to opportunistic message delivery in PSNs. The algorithm will also be shown to not suffer from the premature clustering problem in any of the datasets tested.

## 6.1 Distributed spatio-temporal cluster detection

As well as the distributed spatio-temporal cluster detection algorithm proposed here, there is another algorithm in the literature that does not suffer from either monotonic cluster growth, or the wandering cluster problem.

Borgia et al. [Borgia et al., 2011] proposed a temporal adaptation to the Simple distributed clustering algorithm [Hui et al., 2007b] which can be used to inhibit local cluster growth. Their proposal called AD-Simple relies on pruning clusters of obsolete members using a timer which counts down from the moment PMWDs are entered into local clusters. However, AD-Simple maintains "home clusters" to which nomads and social PMWDs from different time periods are added before being removed once they have not been seen for some time [Borgia et al., 2011]. The time before PMWDs are removed from home clusters is dependent on their cumulative encounter duration, so it is possible that home clusters can retain knowledge of PMWDs that have not been seen in a long time. As a result AD-Simple is not suitable as a means of delivering messages in PSNs as it can suffer from obsolete cluster membership persisting for long periods.

The algorithm proposed in this chapter for distributed spatio-temporal clustering is called Distributed Expectation-Based Spatio-Temporal (DEBT) clustering. DEBT is used to quickly detect spatio-temporal clusters using an expectation-based algorithm similar to SEBS and MEBS cluster detection. Crucially however, DEBT will use shorter time frames, and will quickly remove obsolete PMWDs if they have not been seen for more than one time frame. The restrictions imposed on the DEBT cluster detection algorithm are similar to that of the other distributed clustering algorithms seen in Simple and Nomads:

1. Encounters are asymmetrical.
2. PMWDs can only communicate with other nearby PMWDs during opportunistic pair-wise connections.
3. PMWDs must each keep a record of their own local clusters.

Exactly how clusters are created and maintained in DEBT is addressed in the rest of this section. Opportunistic message delivery protocols for DEBT clusters are considered a separate problem and explored later in Section 6.3.

### 6.1.1 Distributed baseline calculation

DEBT uses the same baseline calculation process for individual PMWDs as is proposed in Section 5.5.2. This provides a mechanism with which to initiate the addition or removal of PMWDs from local clusters without the need to set manual familiar thresholds. Cumulative encounter duration within time frames is again used as this is easy to measure, and allows PMWDs to select others with whom encounters are most predominant despite the parking lot problem [Grasic et al., 2011].

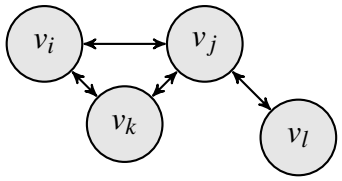
In DEBT clustering, cumulative encounter duration with other PMWDs is continuously assessed against the baseline for the current time frame $t_n$. Baselines for the current time frame are renewed by each PMWD at the end of the previous time frame. For example, a PMWD $v_i$ calculates the mean cumulative encounter duration, $m_i^{t_{n-1}}$ from all encounters during the last complete time frame $t_{n-1}$. $m_i^{n-1}$ is then averaged with the result from the previous $w$ complete time frames to produce the baseline for PMWD $v_i$ for the current time frame $t_n$, referred to as $b_i^{t_n}$. The baseline calculation (which is summarised again in Equation 6.1) is the same calculation which is used for SEBS and MEBS clustering (Equation 5.2). The only difference is that each vertex is calculating the baseline individually in a distributed system.

$$b_i^{t_n} = \frac{m_i^{t_{(n-1)}} + m_i^{t_{(n-2)}} + \cdots + m_i^{t_{(n-w)}}}{w} \tag{6.1}$$

Once the expected baseline for the current time frame has been calculated it can be compared with the cumulative encounter duration for individual PMWDs within the current time frame. Unlike SEBS cluster detection, DEBT has no strongly connected subgraph constraint, and can therefore monitor cumulative encounter duration with

| Neighbours | Branch |
|---|---|
| $v_j$ | $v_i, v_k, v_l$ |
| $v_k$ | $v_i, v_j$ |

(a) Local cluster table for $v_i$




(b) Local cluster of $v_i$

Figure 6.1: Local cluster table for $v_i$, and the corresponding local cluster.

other PMWDs continuously, and act to include nearby PMWDs in local clusters as soon as their cumulative encounter duration reaches the baseline.

For $v_i$ to consider $v_j$ for inclusion in $v_i$'s local cluster, the cumulative encounter duration $x$ between $v_i$ and $v_j$ within time frame $t_n$ (denoted $x^{t_n}_{v_i v_j}$) should be higher than a coefficient, $g_{up}$ multiplied by the baseline for $v_i$ as shown in Equation 6.2.

The parameters $g_{up}$ and $g_{down}$ are introduced in DEBT as a mechanism with which to control cluster size. For example, if the new baseline on PMWD $v_i$ is 20, and the cumulative encounter duration between $v_i$ and $v_j$ in time frame $t_n$ is 30. Then with $g_{up} = 1$ Equation 6.2 is true and $v_j$ will be added to $v_i$'s local cluster. However, if $g_{up} = 2$ then using the same example Equation 6.2 will be false, and $v_j$ will not be included.

$$x^{t_n}_{v_i v_j} > (b^{t_n}_i * g_{up}) \tag{6.2}$$

### 6.1.2 Adding PMWDs to DEBT clusters

Each PMWD running DEBT keeps a *local cluster table* in memory in place of a homogeneous set like cluster. The key components of local cluster tables as shown in Figure 6.1a are the *neighbour* and *branch* columns. The neighbour column is reserved for PMWDs that the local PMWD has met and promoted to the "local cluster". Whilst the branch column is used to store a copy of each neighbour's local cluster table. Each cell in the branch column is usually stored as a set, but a different approach will be proposed later in Section 6.3 that helps to prevent routing loops.

When PMWDs running DEBT connect they exchange local cluster tables, and the information is entered into the local cluster table on the local PMWD once Equation 6.2 has been evaluated to true. Simulated PMWDs running DEBT also request the newest version of the remote PMWD's local cluster table once Equation 6.2 has been satisfied to ensure that data stored locally is up to date. If that request fails then the newest received copy of the remote PMWD's local cluster table is used, and the local PMWD will wait until the end of the frame before requesting the information once more in order to conserve energy and bandwidth for data transmission.

With neighbour and branch data stored in local cluster tables, each PMWD has a view of a cluster to which they belong which may be multiple hops in size as illustrated in Figure 6.1b. However, wireless connections in PSNs are unreliable because of human movement patterns, limited transmission range, and the parking lot problem.

Therefore, data connections to the PMWDs in the local cluster table cannot be guaranteed, but connections to neighbours can be assumed to be more stable than those indicated by the branch data. Furthermore, PMWDs in the branch column are not unique. A PMWD can belong to many branch fields depending on the ordering of encounters and exchanges of local cluster tables, a property which will later be used in Section 6.3.2 to detect paths which contain routing loops.

### 6.1.3 DEBT cluster maintenance

Without a mechanism to remove obsolete PMWDs from local cluster tables, DEBT would suffer from monotonically increasing cluster sizes as seen in many existing distributed clustering algorithms. A number of issues emerged when deciding what mechanism DEBT should adopt in order to remove obsolete cluster members:

1. Removing PMWDs at the end of frames when cumulative encounter duration has fallen below the baseline keeps local cluster tables relevant to the previous time frame, but means that cluster membership may not accurately reflect periods longer than the time frame length.

2. Not removing PMWDs often enough (or at all in the case of home clusters) can cause large message delivery overheads as messages are relayed down obsolete paths.

3. Due to the possible long wait between encounters, PMWDs should be able to act independently in order to remove neighbours. However, cooperation between PMWDs when possible may help to remove obsolete cluster members more quickly. PMWD cooperation will be explored in more detail in Section 7.2.2 in the next chapter.

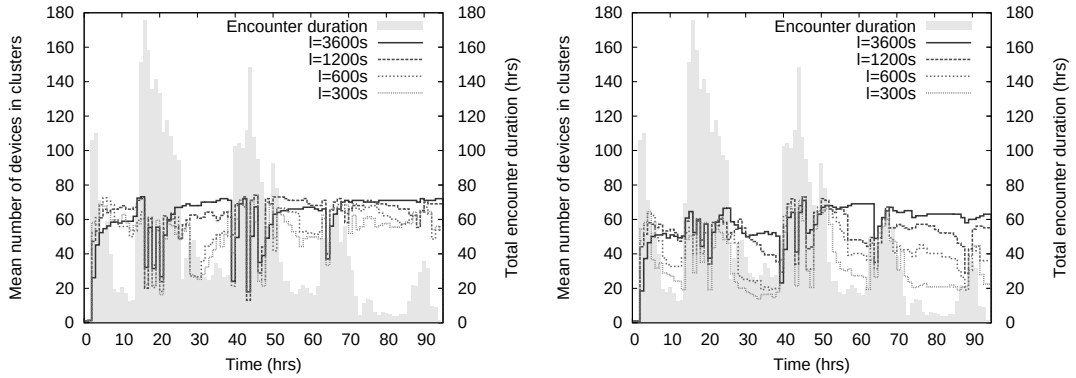
To guard against the first two concerns, remote PMWDs which have failed the test in Equation 6.3 and are present in the local cluster table are not deleted immediately in DEBT, but are marked for deletion at the end of the next frame. Remote PMWDs who are marked for deletion will only be removed from the local cluster table and their corresponding branch data deleted if Equation 6.2 is not satisfied by the end of the next time frame.

$$x_{v_i v_j}^{t_n} \leq (b_i^{t_n} * g_{down}) \tag{6.3}$$

### 6.1.4 Time frame lengths in DEBT

Unlike SEBS cluster detection, DEBT clustering does not need to wait until strongly connected subgraphs are formed. This allows time frame lengths in DEBT to be much shorter than in SEBS cluster detection.

The results of experimentation with different time frame lengths in DEBT cluster detection will be presented in Section 6.2. Time frame lengths $l$ between 300 and 3600




(a) $g_{up} = 1$ and $g_{down} = 1$ (b) $g_{up} = 2$ and $g_{down} = 1$

Figure 6.2: Number of PMWDs in clusters over time in the Infocom6 dataset.

seconds are chosen because there is a high chance that each PMWD will register at least one encounter within this time [Chaintreau et al., 2005].

## 6.2 DEBT cluster analysis

Having a mechanism to delete obsolete cluster members prevents the monster and wandering clusters seen in epidemic label propagation methods. Moreover, DEBT clusters are non-monotonic, and the premature cluster problem is mitigated by using short time frames and baseline calculation over periods that are far shorter than the burst cycle.

The coefficients $g_{up}$ and $g_{down}$ have an effect on how reactive the DEBT protocol is to changing encounter patterns. In the case of the Infocom6 dataset a $g_{up}$ of 1 is too lenient. In other words, as soon as PMWDs are removed from local cluster tables at the end of one frame, they are likely to be re-added to local cluster tables in the next frame. Instead, setting $g_{up}$ to 2 creates new cluster formation patterns that correspond more closely to hourly cumulative encounter durations as the difference between Figure 6.2a and Figure 6.2b shows.

Identifying suitable time frame lengths is critical to matching the rate of change in network topology [Sulo et al., 2010]. An illustration of the effect of time frame lengths $l$ of 300, 600, 1200, and 3600 seconds on the ability of DEBT to match changes in the data is also illustrated in Figure 6.2 using the Infocom6 dataset. In this example, longer values of $l$ lessen the protocol's ability to react to changing network conditions; as shown by a fairly constant number of PMWDs present in local cluster tables when $l = 3600$ seconds. In contrast, DEBT using time frames of 300 seconds is more reactive, and local cluster table membership better reflects connectivity within hourly time frames as shown by the number of PMWDs in local cluster tables in Figure 6.2b.

## 6.3 Message delivery protocols for DEBT clusters

This section details three opportunistic message delivery algorithms for inter and intra-cluster message forwarding via DEBT local cluster tables. The first two protocols DEBT Epidemic (DEBTE) and DEBT Clustering (DEBTC) look at the branch data within the local cluster tables as non-hierarchical sets. The third method called DEBT Trees (DEBTT), preserves path information within the branch data using tree data structures, and offers the ability to spot routing loops at the expense of some additional processing.

### 6.3.1 Epidemic based DEBT forwarding

Epidemic based DEBT (DEBTE) makes use of the neighbour and branch data in message forwarding decisions. If a message destination is anywhere in the local cluster table of an encountered PMWD, then the message will be copied to the encountered PMWD. Using Figure. 6.1 as an example, if $v_i$ has a message for $v_l$ during an encounter with $v_j$, $v_i$ will copy the message to $v_j$. It does this because $v_j$ (the encountered PMWD) has $v_l$ (the destination) somewhere in its local cluster table, as shown in the local cluster table of $v_i$ (the transmitting PMWD).

The forwarding mechanism for DEBTE may result in many duplicate messages being created and forwarded in DTNs, but less so than Nomads where clusters are larger and monotonic. As a possible solution to the overhead problem seen when duplicating copies of a message among cluster members, a second message delivery method called Cluster based DEBT (DEBTC) is proposed.

### 6.3.2 Cluster based DEBT forwarding

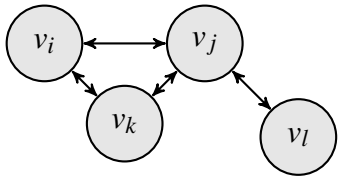
Cluster based DEBT (DEBTC) makes more conservative decisions regarding the message delivery compared with DEBTE in order to lower the number of duplicate messages. A message will not be forwarded to an encountered PMWD if the encountered PMWD also has the transmitting PMWD in the same row of its local cluster table as the destination.

Using the same example as DEBTE, and with help of Figure. 6.1, a message held by $v_i$ and which is meant for $v_l$ will not be copied to $v_j$. This is because $v_j$ has $v_l$ and $v_i$ in the same row of the local cluster table.

Another reason for restricting message duplication between $v_i$ and $v_j$ in Figure. 6.1 is that $v_j$ (the encountered PMWD) may have got its information about $v_l$ (the destination) from $v_i$ (the transmitting PMWD). Therefore, in order to stop routing loops the message is not transferred. Section 6.4 will show that preventing routing loops in this way also results in fewer messages delivered. Therefore, the final message delivery mechanism proposed in this chapter attempts to preserve path information in the local cluster tables using tree data structures, and is called Tree based DEBT (DEBTT).

| Neighbours | Branch |
|---|---|
| $v_j$ | $v_j \rightarrow v_k$<br>$v_j \rightarrow v_l$ |
| $v_k$ | $v_k \rightarrow v_j \rightarrow v_l$ |

(a) Local cluster table for $v_i$




(b) Local cluster of $v_i$

Figure 6.3: An example local cluster table for $v_i$ using DEBTT and image showing the corresponding cluster.

### 6.3.3 Tree based DEBT forwarding

The aim of Tree based DEBT (DEBTT) is to address the case from the example of DEBTC where there may be a path to the destination contained within the branch data, but a message is not transferred because there may be a routing loop. Routing loops cannot be accurately identified in the branch data if the information is processed as a non-hierarchical set of PMWDs, so DEBTT constructs a tree in the branch field by preserving who received local cluster tables from whom.

Again, using the example of $v_i$ attempting to send a message to $v_l$ via $v_j$, but this time using the local cluster table (and enclosed trees) shown in Figure 6.3. A path exists ($v_j \rightarrow v_l$) in the branch column of $v_j$ which does not contain a routing loop and the message will be copied from $v_i$ to $v_j$.

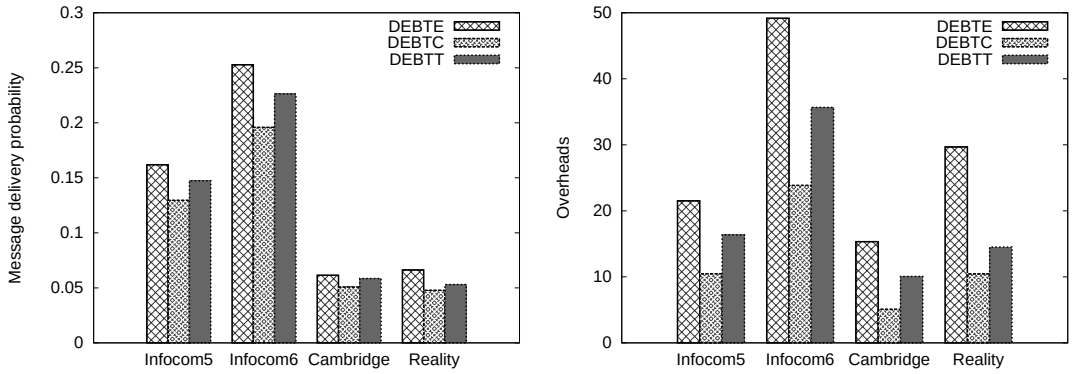
Local cluster tables in DEBTT are trees, but the terminology is kept the same as the other methods for ease of comparison. As local cluster tables in DEBTT are transferred as trees, and added to other trees, a number of implementation measures are needed so that trees do not grow indefinitely.

1. A PMWD will only transmit the shortest paths for individual destinations found in each branch cell. In cases where there are two shortest paths of the same length to the same destination in the same branch cell, both are transmitted (an extended breadth first search is used to do this).

2. Upon receiving a DEBTT local cluster table, the receiver will remove any paths in branch cells that come after itself before inserting the information into its own local cluster table.

## 6.4 Message delivery results for DEBT

In this section the message delivery methods proposed for DEBT are compared against 2 other distributed clustering and delivery methods, Bubble [Hui et al., 2007a] and Nomads. For Bubble, both the K-clique and Simple [Hui et al., 2007b] clustering techniques are used as a basis for the aggregated monotonic clustering.




(a) Message delivery probability

(b) Message delivery overheads

Figure 6.4: Opportunistic message delivery probability and overheads for DEBT protocols.

### 6.4.1 Experimental environment

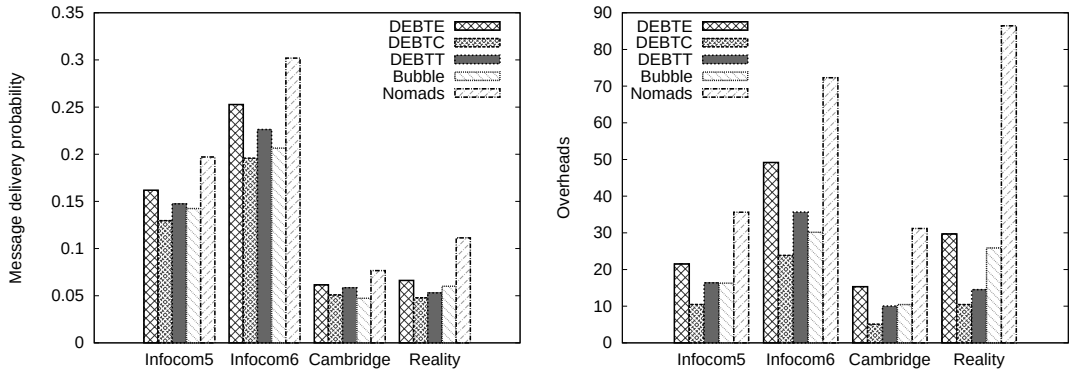
The message generation patterns, communication interface specification, and datasets used for experiments in this chapter are the same as those described in Section 5.7.1 in the previous chapter.

During the design and testing of DEBT it was also found that short frame sizes of 300 seconds are inappropriate for very sparse datasets such as Reality (for dataset density see Section 2.6.1 in Chapter 2). This conclusion was reached experimentally after looking at the cluster composition of the DEBT algorithm over time using frame sizes of 300, 600, 1200, and 3600 seconds. It was discovered that with time frames shorter than 1 hour in experiments involving the Reality dataset that every encounter becomes significant compared to the baseline because there are long sequences of empty time frames between encounters. Furthermore, DEBT cluster tables are not shared with other PMWDs before they are deleted when the time between encounters is more than double the time frame length. Therefore, for the following message delivery results using DEBT, time frames of 1 hour in length are used in Reality and frame sizes of 300 seconds are used for the other datasets.

### 6.4.2 DEBTE vs. DEBTC vs. DEBTT

Figure 6.4 shows the message delivery results for each of the message delivery algorithms proposed for DEBT. The general pattern is that DEBTE will produce more duplicate messages than DEBTC and DEBTT by flooding local clusters until a message reaches its final destination. In contrast, DEBTC will not flood clusters and instead waits for intra-cluster encounters between PMWDs. DEBTT attempts to find a happy medium between DEBTE and DEBTC in terms of message delivery success ratio and efficiency.

It may be reasonable to suggest that DEBTT should deliver the same number of messages as DEBTE, but with fewer duplicate messages. However this is not always




(a) Message delivery probability.

(b) Message delivery overheads.

Figure 6.5: Opportunistic message delivery probability of DEBT protocols compared with Bubble and Nomads.

the case as old paths are still broken unexpectedly and less message duplication results in lower delivery probability in some cases. As a result DEBTT delivered 10% fewer messages than DEBTE overall, but with 34% fewer overheads.

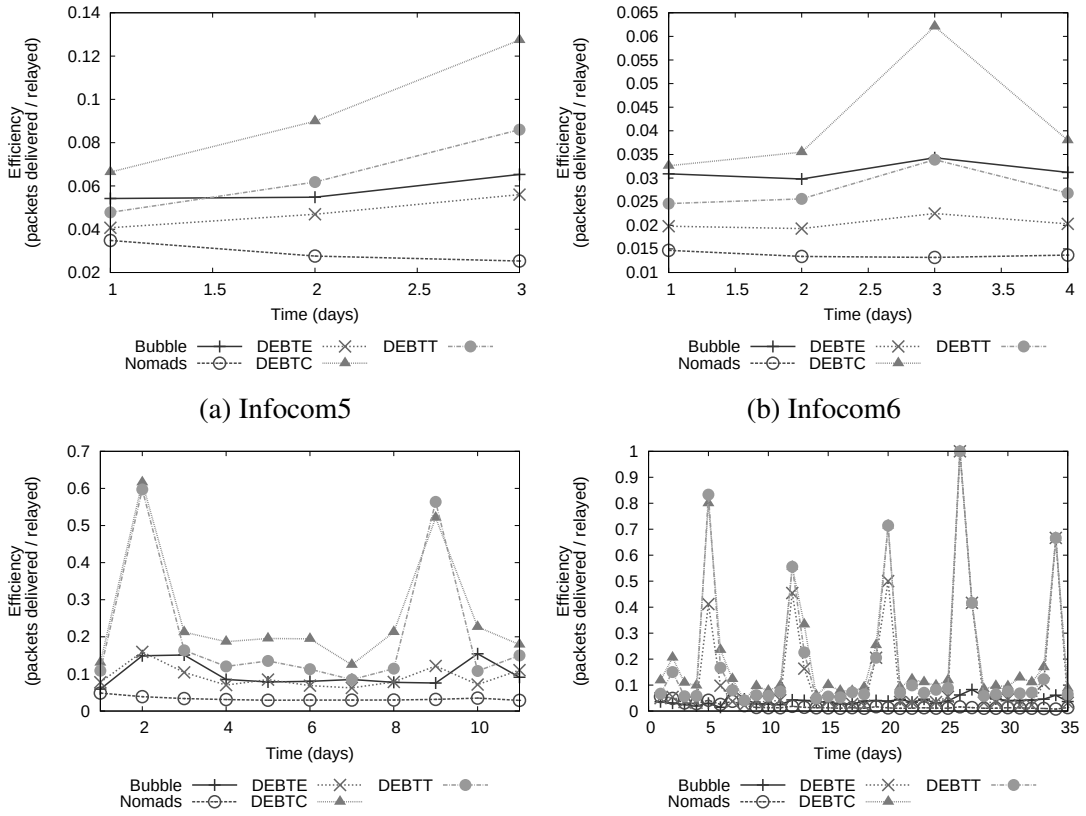
### 6.4.3 Spatio-temporal vs. aggregated monotonic clustering

The purpose of this subsection is to compare the message delivery performance of the distributed spatio-temporal clustering based protocol DEBT, with that of distributed aggregated monotonic clustering based protocols Bubble and Nomads.

The results presented in Figure 6.5a show that message delivery probability of DEBTE, DEBTC, and DEBTT is comparable and often greater than Bubble's. The only dataset where DEBTT does not deliver more messages than Bubble is in Reality, where Bubble delivers 13% more messages but with 80% more overheads because of its monotonic cluster growth. Delivery probability via Nomads still tends to be higher than all of the other methods because Nomads effectively floods networks with duplicate messages during construction of very large clusters, hence the large overheads seen for each dataset in Figure 6.5b.

The efficiency over time for Nomads, DEBT, and Bubble is shown in Figure 6.6. Efficiency in these charts is calculated by dividing the daily number of delivered messages by the daily number of relayed messages. Measured in this way the effectiveness of Nomads is very poor, as large monotonically increasing clusters cause efficiency to diminish over time.

The spikes in efficiency seen in Figure 6.6 are reactions by DEBT to periods of time where there are few encounters and delivery probability is low. During these periods spatio-temporal clusters decrease in size, and fewer messages are relayed. In contrast, the aggregated monotonic clusters used in Bubble and Nomads cause messages to be relayed during the same periods, even when there is very little chance messages will be delivered within the TTL (1 day in Reality and 1 hour in the other datasets).




(a) Infocom5

(b) Infocom6

(c) Cambridge

(d) Reality

Figure 6.6: Message delivery efficiency over time.

The median message delivery results for each protocol are collated in Table 6.1. These results show that the message delivery probability of DEBTE is 21% lower than Nomads, but with half of the overheads. Moreover, the overheads of the DEBTT algorithm are 7% lower than Bubble, with 6% more messages delivered overall.

## 6.5 Summary

This chapter describes a novel distributed cluster detection algorithm called DEBT clustering which can be used to detect expectation-based spatio-temporal clusters. DEBT clusters have been applied to the problem of efficient message delivery in PSNs using three different opportunistic message delivery protocols, and it has been shown in Section 6.4.3 that at least one of these protocols (DEBTT) can deliver 6% more messages with 7% fewer overheads than Bubble.

Unlike the SEBS clusters which are described in Chapter 5, DEBT clusters can be easily detected and maintained using a distributed algorithm. However, automatic detection of time frame lengths may be needed for real world deployment. As a guide, heuristic observations showed that time frame lengths of 3600 seconds in Reality and

| | **Message delivery probability** | **Overheads** |
|---|---|---|
| **Bubble K-Clique** | 0.114 | 20.680 |
| **Bubble Simple** | 0.115 | 20.764 |
| **PRoPHETv2** | 0.145 | 25.683 |
| **Nomads** | 0.172 | 56.389 |
| **DEBTE** | 0.136 | 28.923 |
| **DEBTC** | 0.106 | 12.469 |
| **DEBTT** | 0.121 | 19.146 |

Table 6.1: Mean message delivery results for each protocol across all experiments.

300 seconds for the other tested datasets allow spatio-temporal clusters to grow enough to facilitate effective message dissemination. The reason for this is that when using these numbers there is at least one new encounter per time frame during the bursts described in Section 2.4, but it remains to be seen whether this information can be used to automatically determine time frame lengths in a distributed algorithm, and whether time frames can be adjusted to react to changes in a PMWD's encounter patterns.

It is now becoming increasingly apparent that the choice of clustering algorithm and the way in which clusters are used in forwarding decisions is crucially important to the performance of cluster based forwarding in PSNs. Of the opportunistic message delivery protocols seen so far in this thesis, DEBTT offers the most promising message delivery performance in terms of delivery probability and efficiency. However, the mean message delivery probability reported by DEBTT of 0.121 still falls some way short of the probability of 0.147 given by PRoPHETv2. The next chapter addresses this concern with a new opportunistic message delivery protocol which uses large non-monotonic clusters to deliver messages.

# Chapter 7

# DRAFT clustering

The previous two chapters introduced new methods with which to detect expectation-based spatio-temporal clusters. Chapter 5 detailed two centralised algorithms which can be used to detect expectation-based spatio-temporal clusters in encounter data. Whilst Chapter 6 described DEBT clustering which can be used to detect expectation-based spatio-temporal clusters in a distributed system such as a PSN.

This chapter offers a totally different mechanism with which to detected spatio-temporal clusters, called Distributed Rise And Fall spatio-Temporal (DRAFT) cluster detection. DRAFT has a lot in common with the aggregated monotonic clustering algorithms explored in Chapter 4, with one important difference in that the clusters produced by DRAFT are non-monotonic. The purpose of DRAFT clustering is to detect spatio-temporal clusters that are more suitable for the efficient delivery of large numbers of messages in PSNs than SEBS or DEBT clusters.

## 7.1 Non-monotonic distributed cluster detection

Two of the most advanced opportunistic message delivery algorithms proposed prior to this work are PRoPHETv2 [Grasic et al., 2011] and Bubble [Hui et al., 2007a]. Both have consistently performed well in the simulated PSNs in this thesis, as shown in the message delivery statistics in Table 6.1.

The DEBTT protocol introduced in the previous chapter is capable of delivering more messages, more efficiently, than Bubble in the majority of cases. However, DEBTT is not as effective, nor as efficient, as PRoPHETv2. In this chapter Distributed Rise And Fall spatio-Temporal (DRAFT) clusters are used to deliver more messages, more efficiently, than PRoPHETv2.

DRAFT clusters decay over time in a similar way to AD-Simple, but the algorithm does not preserve home clusters as is described in the next section. DRAFT clusters will also be analysed in Section 7.3, this is followed by the description of a novel opportunistic message delivery algorithm which uses DRAFT clusters in Section 7.4.

## 7.2 Detecting DRAFT clusters

Distributed Rise and Fall spatio-Temporal (DRAFT) clustering is the final distributed spatio-temporal clustering method for PSNs proposed in this thesis. DRAFT clustering involves keeping track of cumulative encounter durations with individual PMWDs, and a decay function similar to that of AD-Simple's so that clusters reflect current and recent encounters by excluding PMWDs which have not been seen for a long time.

The protocol needs up to 3 parameters to be chosen by the user in order to govern the rate at which clusters grow and decay. Suggested values for these parameters are discussed in the following sections and depend on the mobility, expected encounter duration, and what length of time spatio-temporal clusters describe:

1. The familiar threshold of length $\upsilon$ seconds is the threshold at which cumulative encounter durations between PMWDs trigger the cluster inclusion process. The familiar threshold could be substituted for the baseline calculation discussed in the previous chapter, but is static in this chapter for the sake of simplicity.

2. A time frame of length $l$ seconds governs the interval at which the cumulative encounter durations for each PMWD are decayed.

3. The decay ratio $\delta$ which should be in the range $0 \leq \delta \leq 1$ governs by how much cumulative encounter durations are reduced at the end of each time frame.

It is important to point out that after the first time frame, pair-wise encounter durations in DRAFT are no longer truly cumulative. Any cumulative encounter durations greater than zero will be decayed by a certain amount depending on the value of $\delta$. Moreover, it is also important to note that encounter duration decay is multiplicative rather than additive because some PMWDs experience very different encounter durations to others [Grasic et al., 2011]. Therefore, encounter duration decay being multiplicative allows for different levels of connectivity, and the same rate of decay can be specified for the entire network.

### 7.2.1 Building DRAFT clusters

Spatio-temporal clusters in DRAFT are formed opportunistically using the cumulative encounter duration of pair-wise encounters between PMWDs. A PMWD $v_i$ running DRAFT maintains the following three data structures for efficient processing:

1. A set of tuples containing encountered PMWD IDs and associated cumulative encounter durations, called the Neighbour Set $NS_i$.

2. A local spatio-temporal cluster $C_i$.

3. A table $D_i$, containing PMWDs marked for deletion from $C_i$, and some PMWDs that have been deleted.

The process with which DRAFT clusters are then built up can be summarised as:

**DRAFT clustering step 1.** *Initially $C_i$ is set to $\{v_i\}$, $NS_i$ and $D_i$ are set to $\emptyset$.*

**DRAFT clustering step 2.** *When $v_i$ encounters another PMWD called $v_j$, $v_i$ enters $v_j$ into $NS_i$ if it is not already there. $v_i$ then begins to add the duration of the encounter to the record for $v_j$ in $NS_i$.*

**DRAFT clustering step 3.** *If the cumulative encounter duration stored in $NS_i$ for $v_j$ ($NS_{ij}$) exceeds the familiar threshold $\upsilon$, or $v_i$ encounters a PMWD $v_j$ which is already a member of $C_i$, or it is the end of the current time frame on $v_i$ and $NS_{ij} > \upsilon$, then $v_i$ requests $C_j$ and $D_j$ from $v_j$. If the request for cluster data is successful the algorithm then:*

1. $v_i$ adds $v_j$ to $C_i$ if it not already there.
2. If $v_j$ has been marked for deletion by being present in $D_i$ (see Section 7.2.2), then $v_j$ is "forgiven" and removed from $D_i$
3. The algorithm forks to step 5 in Section 7.2.2.

As encounters are asymmetrical, this process is performed independently by all the other PMWDs in the PSN, including $v_j$.

### 7.2.2 DRAFT cluster decay and PMWD cooperation

To facilitate cluster decay, the passage of time is split into a number of discrete time frames of length $l$. At the end of each time frame, associated cumulative encounter durations in neighbour sets are decreased by multiplying them by the decay ratio $\delta$ ($\delta = 1$ no decay, $\delta = 0$ absolute decay).

Cluster membership is reassessed by each PMWD at the end of each time frame, and during encounters between PMWDs:

**DRAFT clustering step 4.** *At the end of each time frame the following checks are performed in order:*

1. Any records in both $C_i$ and $D_i$ are considered old and removed from $C_i$.
2. The records are kept in $D_i$ for commonality tests with other PMWDs, or until the PMWD in the record is added to $C_i$ once more.
3. All connected times for PMWDs in $NS_i$ are multiplied by the decay ratio $\delta$ in order to keep records fresh. Any records which fall below the familiar threshold $\upsilon$ are marked for deletion by being added to $D_i$ ready for the end of the next time frame.

**DRAFT clustering step 5.** *After the DRAFT algorithm forks at the end of step 3, DRAFT also tries to delete obsolete PMWDs:*

1. $v_i$ checks records in $D_i$ against those in $D_j$. As encounters in PSNs are opportunistic, a spatio-temporal commonality test has been passed and any PMWDs which are in both $D_i$ and $D_j$ are deleted immediately from $D_i$ and $C_i$ without waiting until the end of the next time frame.

2. If a record in $D_i$ is in $C_j$ but not $D_j$ then the PMWD is not deleted.

3. If a record is in $D_i$ but not in $C_j$ or $D_j$ then the record is left in $D_i$ in case another PMWD is encountered with a matching record in the future.

To save memory, PMWDs are also removed from neighbour sets once their associated encounter durations decay to below a small number e.g. 0.1 seconds. Informally, the values of $\delta$, $l$, and $\upsilon$ are related to the mobility of the participants and how reactive cluster membership should be, which is determined by the application and/or user. In order to detect only significant relationships, the length of $\upsilon$ should be greater than the mean encounter duration for each PMWD. Whilst $l$ should be at least longer than the minimum inter-encounter time so that there is at least one encounter per time frame.

The combination of $\delta$, $l$, and $\upsilon$ makes the DRAFT algorithm tunable for a variety of different applications. $\upsilon$ can be set manually or by using the baseline calculation from the previous chapter. If clusters are needed which decay quickly, $l$ should be close to the mean inter-encounter time, and $\delta$ should be closer to 0 than 1. Another way to phrase this is that if mobility is high, and clusters should reflect recent encounters, $\delta$ should be low but not zero, and $\upsilon$ close to the mean encounter duration. Conversely, if clusters are needed that reflect longer periods, $\delta$ should be set higher.

### 7.2.3 Message delivery using DRAFT clusters

**Algorithm 6** Message delivery algorithm for DRAFT.

```
for each encountered PMWD as v do
    if DontHaveLocalClusterFrom(v) then
        RequestLocalClusterFrom(v)
    end if
    for each message as m do
        if v=DestinationOf(m) then
            DeliverMessage(v,m)
            DeleteMessage(m)
        else if LocalCluster of v contains DestinationOf(m) then
            CopyAndTransferMessage(v,m)
        end if
    end for
end for
```

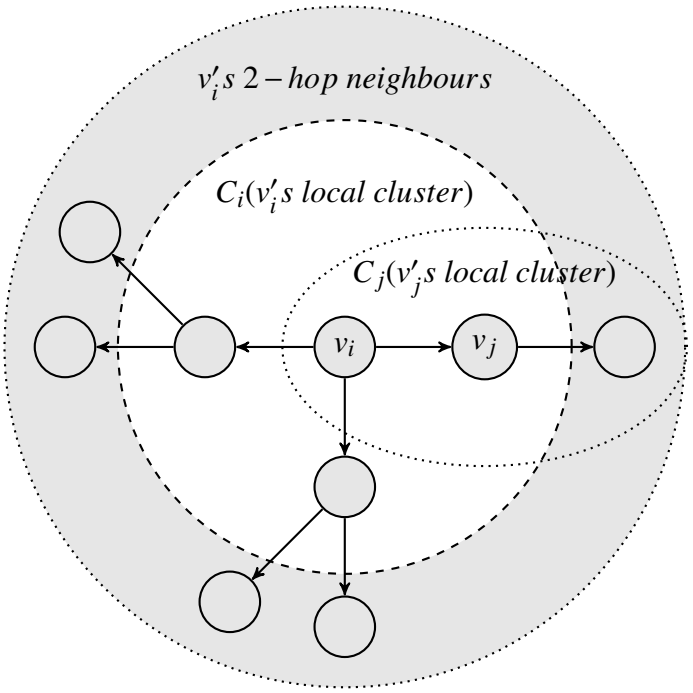



Figure 7.1: A PMWD $v_i$ can see potential 2-hop neighbours upon encountering $v_j$.

The same semi-oblivious message forwarding algorithm that is used in Nomads is adopted for use in DRAFT (see Algorithm 6). This allows for direct comparisons to be made between the message delivery performance of DRAFT clusters against the aggregated monotonic clusters produced by Nomads.

The ability of PMWDs to request local clusters from nearby PMWDs allows them to check for possible 2-hop paths. Note, no explicit extra roles for PMWDs are assigned during this process, and unlike DEBT, information received from neighbouring PMWDs is not stored in local clusters. The possibility of 2-hop opportunistic message routing in DRAFT is simply a consequence of the movement patterns of participants as Section 7.3.3 will show, and being able to either;

1. Ask a remote PMWD if it has a message destination within its local cluster.

2. Or inspect a copy of the remote PMWD's local cluster to see if it contains a message destination.

Both approaches would work with DRAFT, but as PMWDs may have many messages ready to transmit, and the DRAFT algorithm may already have requested a remote PMWD's local cluster, the later approach is used to cut down on the number of requests. The actual checks performed before message duplication are detailed in Algorithm 6. They do not include checks for PMWDs further than 2 hops away as this would require PMWDs to exchange and store a large amount of additional data, which is one of the possible criticisms of DEBT from Chapter 6 as it generated complex local cluster tables depending on its parameters.

| | Simple | | DRAFT | |
|---|---|---|---|---|
| | **Inc.** | **Dec.** | **Inc.** | **Dec.** |
| **Infocom5** | 17.6 | n/a | 46.1 | 17.5 |
| **Infocom6** | 29.8 | n/a | 136.4 | 116.3 |
| **Cambridge** | 10.6 | n/a | 29.5 | 12.5 |
| **Reality** | 16.3 | n/a | 34.4 | 25.3 |

Table 7.1: Mean number of instructions per PMWD issued which increased (Inc) and decreased (Dec) cluster size.

Figure 7.1 illustrates the resulting 2-hop delivery possibilities using example local spatio-temporal clusters of $v_i$ and $v_j$. Upon connecting to $v_j$, the PMWD $v_i$ can inspect $v_j$'s cluster information $C_j$ to see if the destination of a message lies within $C_j$. If it does, then the message can be copied to $v_j$. There is no guarantee that a message will be delivered immediately, or even by PMWD $v_j$ due to PMWD mobility. However, the message is now with both $v_i$ and $v_j$ which may increase the chance of the message reaching the destination without obviously flooding the network.

## 7.3 DRAFT cluster analysis

This section will offer analysis of the DRAFT clusters and possible 2-hop neighbours. It will begin by looking at the size of DRAFT clusters before offering analysis on the time PMWDs spend within DRAFT clusters.

### 7.3.1 Updates to clusters

In both aggregated monotonic and spatio-temporal clustering, clusters undergo a number of changes as they are created. Table 7.1 shows the mean number of changes to spatio-temporal clusters per PMWD in DRAFT with $\delta = 0.8, \upsilon = 120$ seconds, and $l = 3600$ seconds, compared to the aggregated monotonic clustering method Simple [Hui et al., 2007b]. The purpose of this table is just to highlight that DRAFT creates many more cluster formation related operations than an algorithm which can only increase cluster size. This trade off between complexity and message delivery efficiency should be explored in more detail in future work. It is being left out of this thesis as the capabilities of target device and wireless protocol (such as the as yet to be finalised 802.15.8) should be specified in order to create a more meaningful test.

### 7.3.2 DRAFT cluster size

As cluster membership in DRAFT ($\delta = 0.8$, $\upsilon = 120$ seconds, and $l = 3600$ seconds) is continuously reassessed, cluster size varies over time. Table 7.2 shows a comparison

| | Simple | DRAFT |
|---|---|---|
| **Infocom5** | 81.46% | 75.20% |
| **Infocom6** | 79.59% | 49.99% |
| **Cambridge** | 86.81% | 54.37% |
| **Reality** | 55.84% | 13.62% |

Table 7.2: Mean local cluster size as a percentage of dataset size at the end of experiments involving Simple and DRAFT.

between DRAFT and Simple clusters that illustrates the fact that DRAFT clusters are usually smaller than the aggregated monotonic clusters produced by Simple.

Chapter 4 shows that smaller clusters can have a negative impact on message delivery probability because smaller clusters have a smaller chance of containing a message destination. However, the next subsection will show that almost all of the PMWDs in the datasets can be seen when considering 2-hop neighbours in spatio-temporal clusters, even with smaller local cluster sizes.

### 7.3.3 Cluster size and composition over time

Figure 7.2 shows daily snapshots of the mean number of PMWDs contained in local DRAFT clusters and the mean number of possible 2-hop neighbours for each dataset (DRAFT with $\delta = 0.8$, $\upsilon = 120$ seconds, and $l = 3600$ seconds). In the Infocom and Cambridge datasets, the number of 2-hop neighbours is on average 87% greater than the number of PMWDs in local spatio-temporal clusters. In the Reality dataset the proportion of 2-hop neighbours to local spatio-temporal cluster size is much larger, with numbers of 2-hop neighbours being three times the number of PMWDs in spatio-temporal clusters on average.

It is also interesting to note that the number of 2-hop neighbours increases over time, despite local spatio-temporal cluster size decreasing or stabilising over the same period. Furthermore, the set of 2-hop neighbours usually contains most of the PMWDs in each experiment ($> 60\%$ of PMWDs in Reality, and $> 90\%$ in other experiments).

Due to the high number of 2-hop neighbours seen when analysing DRAFT clusters it is reasonable to suggest that the probability of an encountered PMWD having a message destination within its local cluster is high. Or there is at least around a 60% chance that an encountered PMWD has a message destination in its local cluster.

### 7.3.4 Cluster size and 2-hop neighbours

Heat-maps for normalised DRAFT cluster sizes and 2-hop neighbours for hourly snapshots are shown in Figure 7.3. They show a marked difference between Reality and other datasets in terms of both spatio-temporal cluster sizes and 2-hop neighbours.

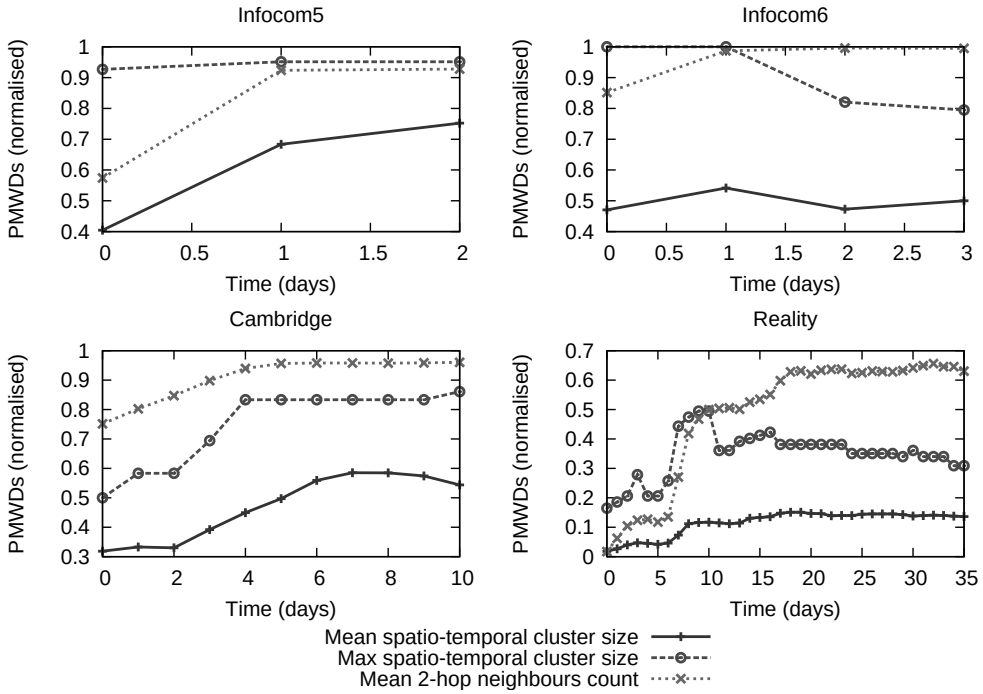


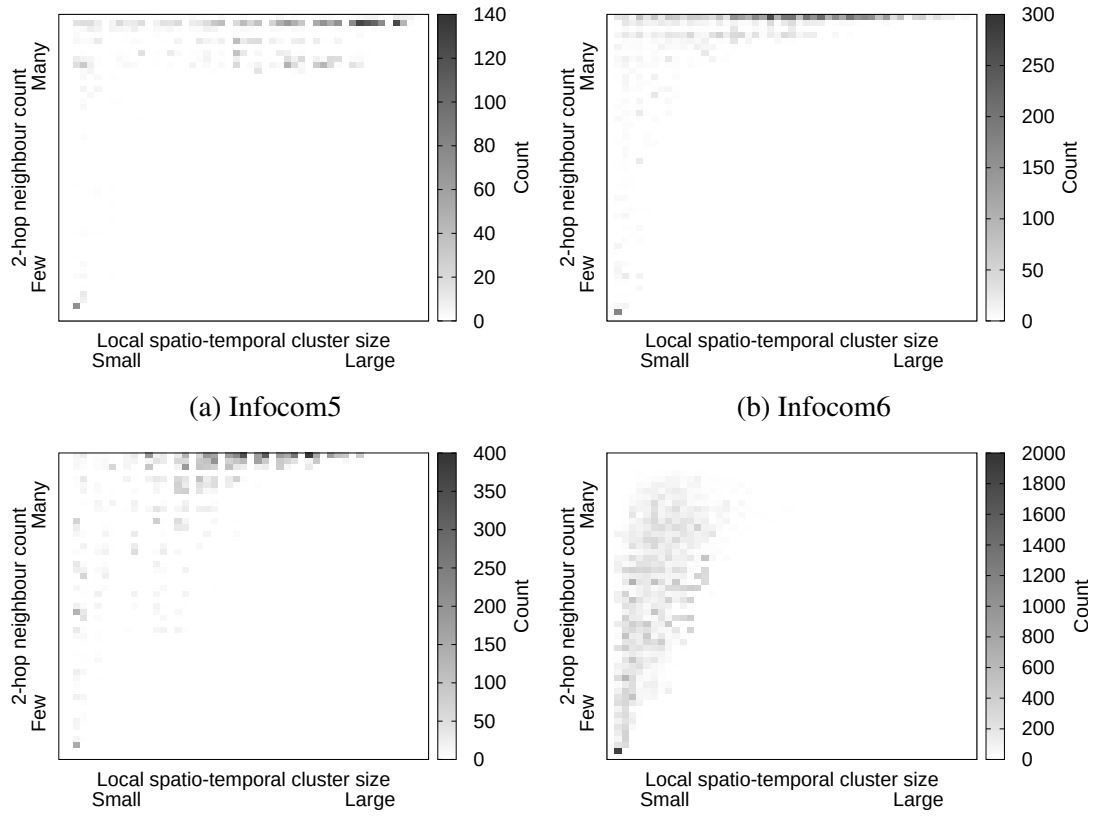

Figure 7.2: Mean and max local spatio-temporal cluster sizes, and 2-hop neighbours.

In the Reality dataset there are few occasions when the numbers of potential 2-hop neighbours is greater than 60% of the dataset size, and almost no occasions where the number of 2-hop neighbours is as high as 80%. Moreover, DRAFT cluster size in Reality remains low with a mean hourly cluster size of just 10 PMWDs compared with 16 in Cambridge, 25 in Infocom5, and 38 PMWDs in Infocom6. Later, Section 7.4.3 will show that the small cluster size and number of 2-hop neighbours in Reality will have a negative effect on DRAFT message delivery.

### 7.3.5 Time spent in spatio-temporal clusters

Chapter 5 included a discussion on how time spent in spatio-temporal clusters is dependent on how the spatio-temporal clusters themselves are defined, and what parameter values are used to detect them. DRAFT is no different, and spatio-temporal cluster membership times depend on factors such as decay rate, familiar threshold and time frame size. Figure 7.4 shows the probability that spatio-temporal cluster membership time will exceed a given value $x$ in each dataset for $\delta$ values of 0.1,0.5, and 0.8 when using a time frame length of 1 hour and a familiar threshold of 120 seconds.

Interestingly, the data shows that with a fast decay rate $\delta = 0.1$, over 81% of all cluster memberships still last longer than 1 hour. In Reality though, 98% of spatio-temporal clusters memberships last longer than an hour, suggesting that the number of nomads in the Reality dataset is low.




(a) Infocom5

(b) Infocom6

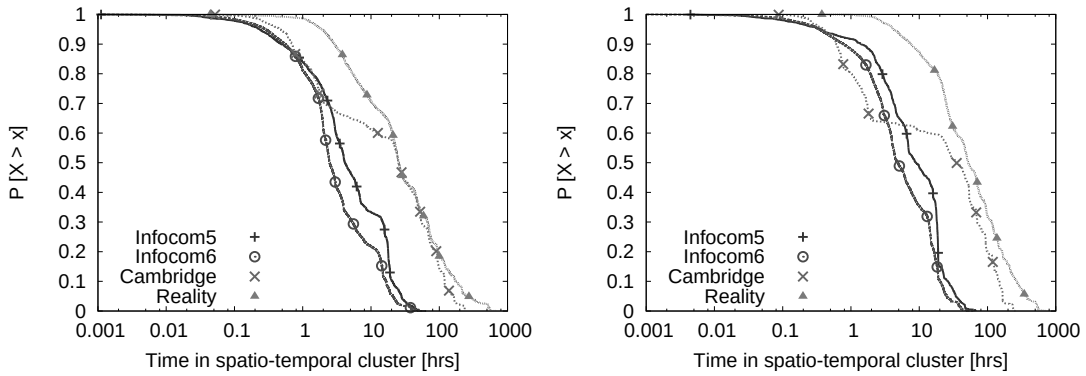
(c) Cambridge

(d) Reality

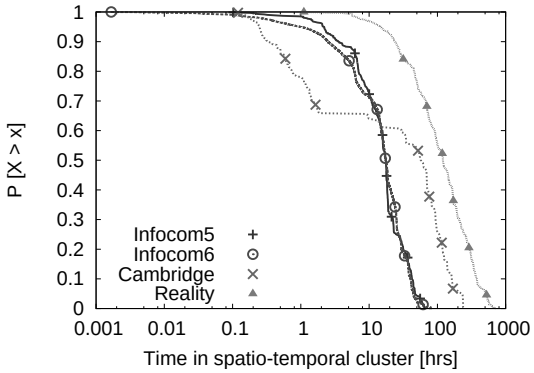
Figure 7.3: High resolution heat map showing the number of 2-hop neighbours against the size of the spatio-temporal clusters produced by DRAFT. The count is the number of times a certain combination was seen in each dataset.

There is also a sizeable difference in DRAFT cluster membership duration between the conference and campus environments. In the Infocom5 and Infocom6 datasets 40% and 57% of PMWDs spend longer than 3 hours in DRAFT clusters respectively. However, the time spent in DRAFT clusters is generally less in the campus datasets. In the Cambridge dataset less than 30% of PMWDs spent longer than 3 hours in DRAFT clusters before being removed, and in Reality the number is only 10%. This indicates that the campus wide experiments have a more diverse selection of participants who interact for shorter periods than participants at a conference.

If the findings about the time spent in DRAFT spatio-temporal clusters are combined with the size of DRAFT clusters from Section 7.3.3, a rich picture emerges of the types of DRAFT clusters seen in the datasets. Large DRAFT clusters are more common in the conference datasets, and PMWDs also spend more time in DRAFT clusters at conferences than at a university campus. This is intuitive as conferences are often held in confined spaces. So it is reasonable to suggest that the cause for this behaviour is that some participants spend more time in the proximity of others at a conference than they would during their working day.




(a) $\delta = 0.1$ (b) $\delta = 0.5$




(c) $\delta = 0.8$

Figure 7.4: Cumulative probability distribution of cluster membership times.

## 7.4 Message delivery performance of DRAFT clusters

In this section the DRAFT protocol is compared against two opportunistic message delivery protocols which utilise aggregated monotonic clustering, Bubble and Nomads. The performance of DRAFT will also be compared against PRoPHETv2 which does not use clusters in message forwarding decisions, but provides state of the art delivery probability and efficiency. For Bubble, both the K-clique and Simple clustering techniques are used to provide the aggregated monotonic clusters needed for the experiments.

### 7.4.1 Experimental environment

The opportunistic message delivery experiments of this chapter are again conducted using the ONE simulator as described in Section 5.7.1. Even though the simulations assume stable bi-directional data connection during encounters, the DRAFT algorithm does not break down if requests for information fail. In fact, failure to reply to requests is handled in the DRAFT protocol, and this would help to prune the epidemic distribution tree of unreliable links.

| Method | Message delivery probability | Overheads |
|---|---|---|
| **Bubble** | 0.114 | 20.697 |
| **PRoPHETv2** | 0.145 | 25.683 |
| **Nomads** | 0.172 | 56.389 |
| **DRAFT ($\delta > 0.5$, $\upsilon = 120$ seconds)** | 0.147 | 25.348 |
| **DRAFT ($\delta = 0.99$, $\upsilon = 5$ seconds)** | 0.162 | 37.188 |

Table 7.3: Mean message delivery results across all experiments. Time frame length for DRAFT is always $l = 3600s$.

### 7.4.2 Overall results

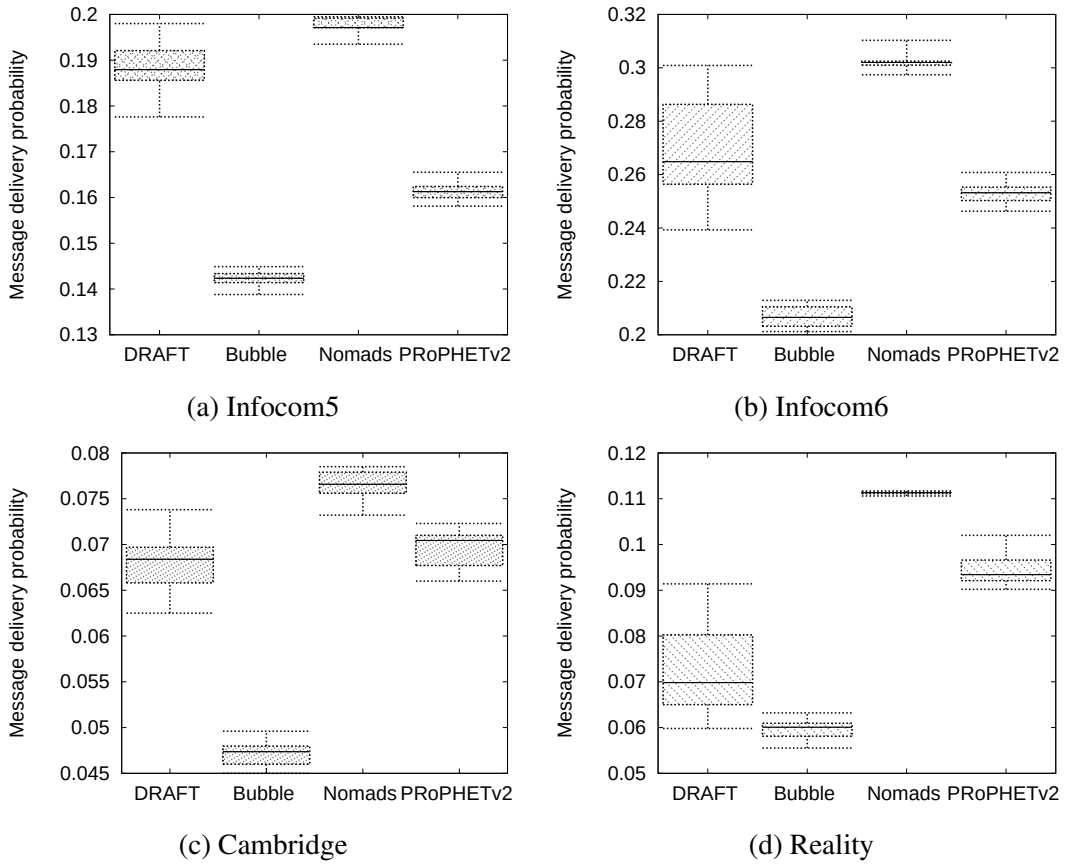
When using $\delta$ values less than 0.5 and when $l = 3600s$, DRAFT's spatio-temporal clusters decay too rapidly to be used in opportunistic message delivery. With $\delta$ values in the range $0.5 < \delta \leq 0.99$, $\upsilon = 120$ seconds, and $l = 3600$ seconds, Table 7.3 shows that DRAFT offers a small performance increase over PRoPHETv2 with 1% more messages delivered with 1% fewer overheads.

The variation of the opportunistic message delivery results for each protocol can be found in Figure 7.5. Generally, DRAFT can be counted upon to deliver more messages successfully than either Bubble or PRoPHETv2, but not Nomads. Nomads is still the most effective protocol tested in terms of message delivery probability because using large aggregate monotonic clusters as a guide for message duplication causes more duplicate messages than when using spatio-temporal clusters (as we saw from the overheads of Nomads compared to opportunistic message delivery using SEBS and DEBT clusters in chapters 5 and 6 respectively).

Figure 7.5 also shows that DRAFT's message delivery probability is comparatively lower than that obtained using PRoPHETv2 in the Reality case. A reason for this low delivery rate is likely to be the small size of spatio-temporal clusters as seen previously in Figure 7.2 and Section 7.3.4. This hypothesis is explored in Section 7.4.3.

### 7.4.3 Opportunistic message delivery in the Reality dataset

One mechanism to increase average spatio-temporal cluster size in DRAFT is to lower the rate of cluster decay. For example, at $\delta = 0.99$ encounter durations are only decreased by 1% at the end of each time frame. Also, by lowering the familiar threshold $\upsilon$, more PMWDs are included in local spatio-temporal clusters in the first instance. If $l$ still equals $3600s$ but $\upsilon$ is limited to 5 seconds and $\delta$ set to 0.99, the resulting clusters are 3 times larger in Reality than when $\delta = 0.8$, $\upsilon = 120$ seconds, and $l = 3600$ seconds (see Figure 7.6a), but message delivery probability in the Reality dataset using these settings is still 6% lower than that given by PRoPHETv2. The trade off is efficiency, with PRoPHETv2 needing to relay twice as many messages as DRAFT to achieve a slightly higher message delivery probability as illustrated in Figure 7.6b.




(a) Infocom5

(b) Infocom6

(c) Cambridge

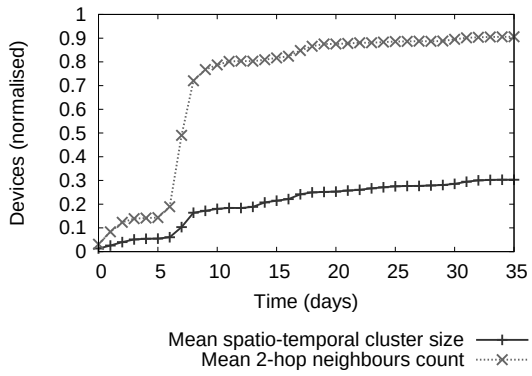
(d) Reality

Figure 7.5: Minimum, first quartile, median, third quartile and maximum message delivery probability for each dataset.

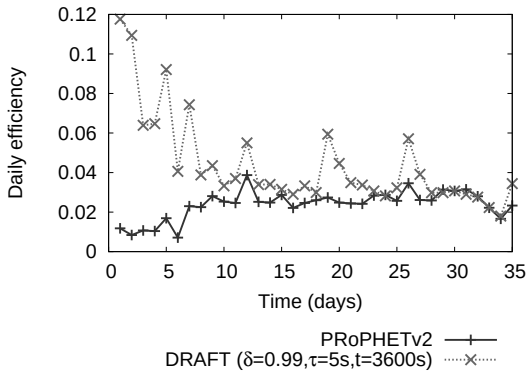
Almost all of the efficiency gains of DRAFT over PRoPHETv2 shown in Figure 7.6b come at the start of the experiment. Further inspection of Figure 7.6a shows that spatio-temporal clusters during the early stages of the experiment are still very small compared with later on. Moreover, Figure 7.6c shows that DRAFT is delivering slightly fewer messages than PRoPHETv2 during this period for reasons which appear to be related to cluster size. As the early stages of the Reality experiment have low delivery success rates, and local spatio-temporal cluster sizes are lower than 25% of the total data set size, the findings are consistent with those in [Orlinski and Filer, 2012c] which suggested that mean cluster size should be 33% of the total dataset size in order to reliably disseminate the maximum amount of data using this 2-hop delivery method.

### 7.4.4 Efficiency over time

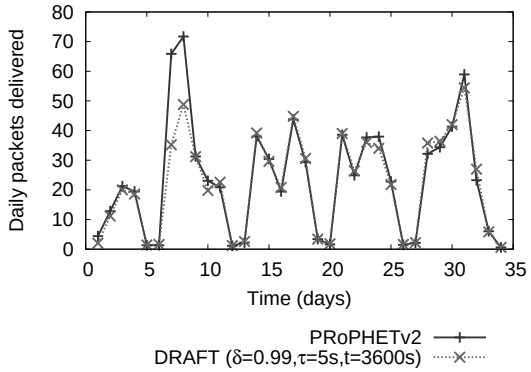
One of the predictions for spatio-temporal clustering is that it will improve the efficiency of message delivery algorithms which currently make use of aggregated monotonic clustering. This hypothesis in now explored again using DRAFT clusters.




(a) Cluster composition.




(b) Efficiency.




(c) Message delivery.

Figure 7.6: There are often long periods of time between encounters in the Reality datasets. As such decay rate and familiar thresholds have been altered for these results to be $\delta = 0.99$ and $\upsilon = 5$s respectively ($l$ is still $3600s$).

The message delivery mechanism in DRAFT is the same as in Nomads. Even so, Figure 7.7 shows DRAFT is a marked improvement on the data-delivery efficiency of Nomads across each of the datasets explored. Therefore it is fair to say that spatio-temporal clustering is more efficient as it performs better than Nomads in terms of creating fewer duplicate messages.

Figure 7.5 also shows us that DRAFT does not deliver as many messages to their final destinations as Nomads in the Infocom6 and Reality datasets. To attempt to correct this, $\upsilon$ was limited to 5 seconds and $\delta$ set to 0.99, and all experiments were repeated. However, Table 7.3 shows the results of doing this were that the message delivery probability of DRAFT is still 6% lower than Nomads, but with 34% fewer overheads.

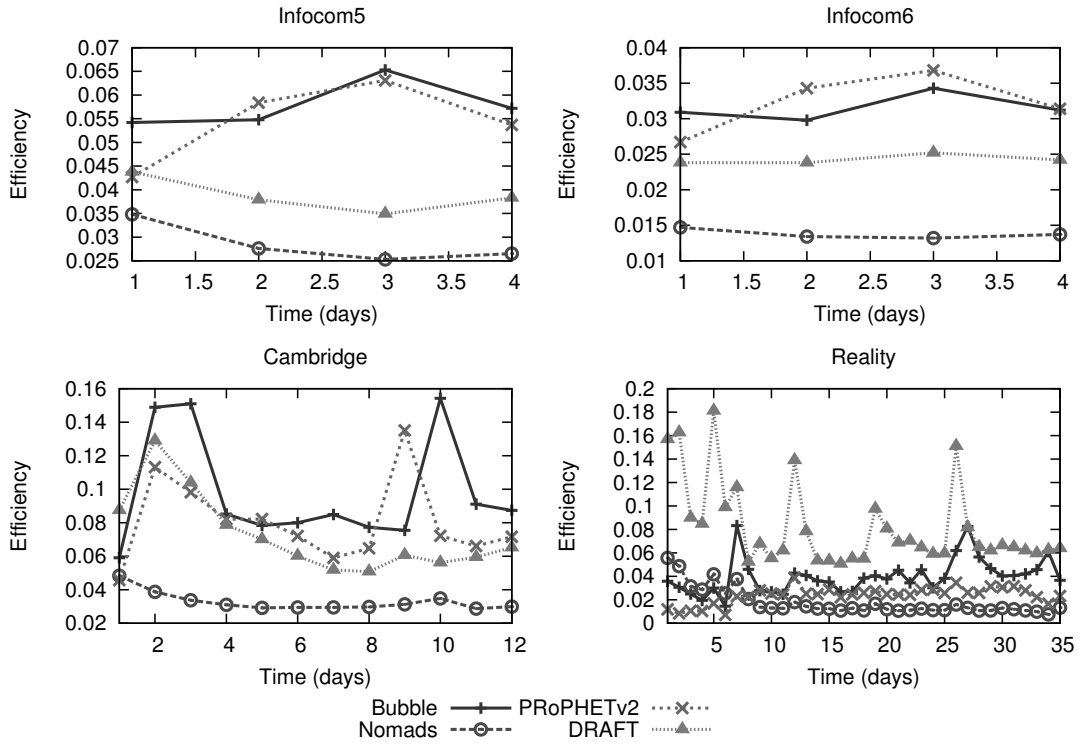



Figure 7.7: Efficiency over time. DRAFT settings, $\delta > 0.5$, $\upsilon = 120$ seconds, and $l = 3600$ seconds

## 7.5 Summary

The findings of this chapter go some way towards satisfying the objectives outlined in Section 1.6.2. By opportunistically forwarding messages using spatio-temporal clusters, the DRAFT protocol has shown that it is possible to efficiently send messages between participants in simulated PSNs.

The efficiency savings seen when using spatio-temporal over the aggregate monotonic clusters of Nomads (sections 6.4.3 and 7.4.4) suggest that correct temporal adjustments are crucial for the efficiency of opportunistic message delivery protocols. However, Figure 7.7 shows that DRAFT is not as efficient as Bubble or PRoPHETv2 in 3 of the 4 tested scenarios. This represents a backwards step in terms of efficiency from DEBTT, which is both more efficient and delivers more messages than Bubble.

In the scenarios where DRAFT is less efficient than Bubble and PRoPHETv2, DRAFT delivered more messages (Figure 7.5). Improvements to the efficiency of DRAFT may be possible whilst keeping the high delivery rates by adopting a hybrid delivery approach similar to Bubble's, or by one of the other methods that are discussed in Section 8.3 of the next chapter.

# Chapter 8

# Conclusions

This thesis presents an investigation into two areas critical to the provision of the PSNs discussed in Section 1.6, autonomous neighbour discovery and opportunistic message delivery. Human movement and encounter patterns are also studied in order to ensure that PSNs work for participants, rather than relying on participants changing their behaviour patterns in some way to meet the needs of the network.

## 8.1 Summary of results

This section summarises the contributions contained within this thesis in the areas of autonomous neighbour discovery, human behavioural analysis, and opportunistic message delivery.

### 8.1.1 Autonomous neighbour discovery for PSNs

Chapter 3 demonstrates that many of the encounters between PMWDs measured with a transmission radius of 10 meters are brief. With that in mind, the first major contribution of this thesis is the discovery that the average hourly out-degree centrality of PMWDs in a PSN is proportional to the number of encounters lasting less than 20 seconds which are detected. Moreover, Section 3.4 shows that many encounters crucial to the level of connectivity in a PSN will be missed if the time between symmetric neighbour discovery processes in a PSN is longer than that given by the IPC equation in Section 3.3.

As the time between the neighbour discovery processes used in reality mining experiments is often much longer than that given by the IPC equation, it is reasonable to assume that many short encounters between PMWDs are missed, and the potential connectivity of the scenarios given in Section 1.7.1 is being understated. Therefore, any method for neighbour discovery in future reality mining experiments should be able to detect nearby PMWDs at a much finer granularity than is currently being used.

The provision of PSNs and other peer aware networks is an emerging discipline, and the means for autonomous neighbour discovery has not been fully standardised. Of the autonomous neighbour discovery protocols tested in Chapter 3, the proposed approach called PISTONS offers the most reliable neighbour discovery results. PISTONS gives a mean hourly out-degree centrality that is 78% higher in fast moving WDM model simulations than DWARF, and over twice the out-degree centrality of STAR.

The performance of PISTONS is linked to the bursty patterns of inter-human encounters seen in Section 2.4. When PISTONS registers an encounter it immediately lowers the inter-probe time of the local PMWD using the IPC equation from Section 3.3, and inter-probe times are increased when neighbour discovery processes are unsuccessful. This behaviour is intentional; the bursty encounter patterns of PMWDs tell us that when an encounter is detected there will likely be more in the near future, and when no nearby PMWDs are discovered during a search it is unlikely that there will be any new encounters until the next burst.

### 8.1.2 Inter-human encounters as graphs

Chapter 5 describes how information relating to inter-human encounters can form strongly connected subgraphs. As well as being useful in the design of clustering and opportunistic message delivery algorithms, these findings offer insights which can be used to validate future human movement models and encounter simulations:

1. Section 5.3 gives evidence that aggregated encounter graphs generated from reality mining datasets densify at a consistent rate, and have a DPL exponent of 1.25.

2. Section 5.6.1 shows that the mean size of the strongly connected components (the largest strongly connected subgraphs) generated in hourly time frames is 5 PMWDs

### 8.1.3 Spatio-temporal cluster detection

The term nomads is used here to categorise PMWDs that move between clusters of social PMWDs [Pietilainen and Diot, 2012]. The nomads encountered by PMWDs are aggregated into clusters in Chapter 4, and these clusters are used in opportunistic message duplication decisions. However, this approach is shown to be unsuitable for opportunistic message delivery in Section 5.1 because of the size of clusters and the numbers of duplicate messages created monotonically increases.

The second major contribution of this thesis is in the development of new spatio-temporal clustering algorithms specifically targeted at detecting when humans congregate in transient groups. These algorithms are tested in chapters 5, 6, and 7 using some available reality mining datasets spanning both conference and campus scenarios.

The most interesting new results in this area were the ones relating to how long participants spend in spatio-temporal clusters and the size of spatio-temporal clusters:

1. Section 5.6.2 describes the MEBS clusters constructed from strongly connected subgraphs detected in hourly time frames. It was shown that MEBS cluster tend to contain 3 PMWDs in the conference data, and 5-6 PMWDs in the experiments conducted at university campuses. It was also discovered that only around 15% of MEBS clusters last longer than 2 hours in all of the datasets tested.

2. Section 7.3.5 describes how long participants spend in DRAFT clusters, which unlike MEBS clusters are detected using a distributed algorithm. These results show that participants spend much longer in spatio-temporal clusters that are not reliant on the formation of strongly connected subgraphs.

### 8.1.4 Opportunistic message delivery via spatio-temporal clusters

In order to meet the objectives outlined in Section 1.6.2, this thesis proposes several new opportunistic message delivery protocols for PSNs that make practical use of spatio-temporal clustering and human movement patterns to deliver messages to their intended recipients. Two notable contributions in this area are:

1. A new distributed cluster detection and opportunistic message delivery protocol called DEBTT is shown to deliver 6% more messages with 7% fewer overheads than Bubble in Chapter 6. DEBTT does this by making use of expectation-based spatio-temporal clusters rather than the aggregated monotonic clusters used in Bubble.

2. Chapter 7 proposes DRAFT clustering which utilises device cooperation as well as a decay function to remove obsolete cluster members. With DRAFT it is demonstrated that opportunistic message delivery that utilises spatio-temporal clustering can be slightly more effective (1% fewer overheads) than the approach used by PRoPHETv2.

Whilst these result are only a small improvement over the current state of the art, there is still plenty of room for improvement. Some suggestions on how to proceed are given in Section 8.3.5.

## 8.2 Critique

The findings contained within this thesis show that there is (as yet) no panacea for opportunistic message delivery in all conceivable PSNs. The message delivery probability and overheads differed greatly between the protocols and datasets tested. This section summarises some of the potential pitfalls that were discovered during this work.

### 8.2.1 Low message delivery probability

The small number of participants and short transmission range of PMWDs means that opportunistic message delivery in simulated PSNs created from existing reality mining datasets can be extremely unreliable. Table 7.3 shows that the maximum opportunistic message delivery probability within 1 hour using a random source and destination is just 0.172. Opportunistic message delivery probability can be increased using the current datasets only if the TTL is raised above 1 hour, or message destinations are carefully chosen to be no more than 1 or 2 hops away. Future work will need to establish if the low delivery probability is a product of low participation in the reality mining experiments or because of some limitation of opportunistic forwarding using human movement patterns.

The current inability to demonstrate that opportunistic message delivery protocols can deliver messages within a short TTL is one of the reasons why PSNs are a likely candidate for DTN technology. Despite this, one should not rule out the prospect of real time applications for PSNs in the future. Perhaps with larger datasets we will see that timely end-to-end communication between participants is possible, or advancements in wireless technology will increase the communication range between PMWDs.

### 8.2.2 High message delivery overheads

A common criticism of multi-copy opportunistic message delivery is that it can create a lot of duplicate messages. Whilst this is true in many cases (such as with Epidemic and Nomads), the comparison between DRAFT and PRoPHETv2 in Section 7.4.2 shows that overheads can be limited with temporal protocols.

Opportunistic message delivery may never be suitable for unicast transmission where there is a reliable end-to-end path, but that is not the problem it hopes to address. Opportunistic methods are being designed to work in situations where the probability of being able to deliver a message via a particular path is difficult to calculate. Therefore, until better route prediction for PSNs is available, it is to be expected that some loss or duplication of messages will occur in response to there being many unreliable paths between PMWDs.

### 8.2.3 Distributed spatio-temporal cluster detection

Cumulative encounter duration within discrete time frames is used extensively in this thesis because of the parking lot problem. However, it may also be possible for distributed algorithms to detect spatio-temporal clusters via other means including:

1. **Velocity.** If PMWDs exchange their current movement speed and direction of travel during encounters (similarly to MoVe [LeBrun et al., 2005]), then the likelihood of PMWDs staying at or travelling to a certain location can be included in clustering decisions.

2. **Previous locations.** Clustering algorithms can be augmented by sharing information relating to the geographic spaces where participants spend a lot of time.

It should also be pointed out that the analysis of MEBS spatio-temporal clusters from Chapter 5 appears to contradict the analysis of DRAFT clusters in Chapter 7. Analysis of MEBS clusters reveals smaller and longer lasting spatio-temporal clusters in experiments conducted at conferences than in those conducted at universities; whilst the analysis of DRAFT clusters reveals larger and longer lasting clusters in the conference datasets. One reason for the different conclusions reached is that DRAFT clustering can add nomads to local clusters; whereas nomads do not necessarily contributed to the strongly connected components that are the fundamental building blocks of MEBS clusters. The differences seen in the analysis of MEBS and DRAFT clustering highlights the importance of understanding the context of the experiment and the inner workings of the spatio-temporal cluster detection algorithm being used before making conclusions about reality mining data.

### 8.2.4 Budget based distributed clustering

Whilst it could have been argued that the results in Sections 4.3.4, 7.3.3, and 7.4.3 suggest clusters generated for the purpose of opportunistic message delivery should contain between 25% and 50% of the encountered PMWDs (10 to 38 PMWDs in datasets tested), this may not be scalable for larger experiments.

Establishing more accurate upper bounds for cluster size will require experiments orders of magnitude larger than the ones conducted in this thesis. However, increasing participation in reality mining experiments requires more time and money than is currently practical, and simulating large numbers of PMWDs communicating over wireless protocols whilst moving to realistic movement patterns is extremely computationally intensive.

In the near future academics will have access to larger human movement and encounter datasets that will enable further research in this area. The telecommunications company Orange have recently given a select group access to fine grained movement traces relating to 5 million mobile phone users in the Ivory Coast.[1] We anxiously await their findings, and hope that this invaluable resource will be freely available soon after.

## 8.3 Future work

Whilst the biggest challenge for new PSN technology may be in the changing of attitudes toward a more open, dynamic, and collaborative Internet – this section focuses on future work in the context of what is contained within this thesis. This section also briefly explores what new collaborative applications are possible with advances in network science, mobile sensing, and access to ubiquitous PMWDs.

[1] Further information about the Orange Data for Development (D4D) challenge can be found on their website http://www.d4d.orange.com

### 8.3.1 Peer aware communications

As well as PISTONS, the IPC method can be used to calculate the inter-probe time for any number of other protocols. Improvements in the out-degree centrality/number of neighbour discovery intervals ratio for PISTONS may be gained in WDM experiments by decreasing $\alpha$ faster after unsuccessful neighbour discovery intervals, or by:

1. Taking into account seasonal encounter information [Wang et al., 2009a].

2. Attempting to predict when bursts of new encounters will occur.

3. Changing base speed, $B$ to be dependent on actual movement speed at the time, detected for example by accelerometers on PMWDs [Bedogni et al., 2012].

4. Contextual adaptation of $\alpha$, i.e., if large numbers of PMWDs are detected without subsequent data being exchanged, then it may be desirable to decrease $\alpha$ to preserve battery life.

In empirical experiments it may be desirable to combine neighbour discovery and the exchange of routing information to save both bandwidth and battery power. For example, neighbour discovery beacons can also contain the information needed by clustering algorithms. However, a more detailed description of the Medium Access Control (MAC) protocol than that given in this thesis will be needed in order to make realistic recommendations.

### 8.3.2 Encounter prediction

In each message delivery protocol analysed in this thesis there is a trade off between message delivery success rates and efficiency. Without better methods of predicting the whereabouts of PMWDs, or knowing the likelihood of there being some interaction with other PMWDs in the near future, it may not be possible to push the efficiency of these protocols further.

The diurnal patterns described by Henderson et al. [Henderson et al., 2008] and the burst cycles seen in Section 2.4 hint that it may be possible to accurately predict human mobility patterns using PMWDs. Existing methods that attempt to predict the whereabouts of people include using a record of previous locations visited or exploiting the mobility data of friends [Scellato et al., 2011b, De Domenico et al., 2012]. However, these approaches are only suitable for predicting the social or regular encounters between PMWDs, which Pietilainen et al. have already shown to have little impact on message dissemination in PSNs [Pietilainen and Diot, 2012].

A challenge for mobility prediction is how to predict encounters involving PMWDs that do not dwell within the same cluster or at the same geographic location for any great length of time.

### 8.3.3 Baseline calculation using forecasting

The baseline calculation for SEBS, MEBS, and DEBT clustering first described in Section 5.5.2 can remove the need to manually configure familiar thresholds. Calculating baselines is just one method of forecasting data points in time series data. It will therefore be interesting to experiment with other forecasting techniques, including the Exponential Smoothing (ES) and the Holt-Winters forecasting methods [Chatfield, 2003] to see their impact on distributed cluster detection. If a more suitable technique can be found then this may also remove the need for the coefficients used in DEBT clustering in Section 6.1.1.

### 8.3.4 Time frames

The discrete time frames which are described in Section 2.5.3 are used throughout chapters 5, 6, and 7. Discretisation offers a convenient way with which to describe changing encounter patterns in reality mining data. However, it may be possible to improve the heuristic approach used to select time frame lengths.

It is not possible to exhaustively test all possible time frame lengths, therefore it will be necessary to experiment with automatic and fuzzy granulation techniques for temporal data to see if they can be used in reality mining [Yu and Cai, 2010].

It should also be possible to devise distributed spatio-temporal clustering algorithms which do not partition time series data into discrete time frames. For example, DRAFT uses the end of time frames to indicate when to decay the sum of encounter durations. In retrospect it may have been possible to replace this with an algorithm that calculates the rate of decay as a function of incoming encounters.

### 8.3.5 Single copy and context aware delivery

Single copy message delivery using clusters is not addressed in this thesis. Instead, flooding-based protocols that spray copies of messages within the boundaries of a cluster are used to compete with the message delivery performance of PRoPHETv2. In secure applications and networks made up of low power devices, it may be desirable to revisit single copy message delivery algorithms such as those described by Spyropoulos et al. [Spyropoulos et al., 2004] and in Message Ferry [Lindeberg et al., 2012].

Of the multi-copy, opportunistic message protocols proposed in this thesis, DRAFT offers the highest message delivery probability. DRAFT can deliver 95% of the messages delivered by Nomads but more efficiently (see Table 7.3). How to deliver the final 5% of messages in a timely and efficient fashion is still an open problem, and may rely on factors other than the encounter history of PMWDs (such as a PMWDs current "context" [Ouchi and Doi, 2012], or the encounter burst prediction covered in Section 8.3.2).

Another important question for future research will be, when is it appropriate to use opportunistic message delivery? In this thesis simulated PSNs are only capable of decentralised ad hoc networking, whereas in a real system the PMWDs may have

access to cellular or Wi-Fi networks some or all of the time. It may be the case that PMWDs can detect congestion in the cellular network and fall back to opportunistic forwarding for non urgent messages, or sending text messages opportunistically when temporarily outside of cell range may prove to be an effective way of extending cell coverage.

### 8.3.6 New collaborative applications

Recently there has been interest from technology companies such as Toshiba in what they call "next generation context aware computing" [Ouchi and Doi, 2012]. The idea of which is to allow mobile phones to sense their surroundings and thus detect the context in which they are being used. For example, by combining readings from accelerometers and gyroscopes, mobile phones can detect users current mode of transportation [Bedogni et al., 2012]. The accuracy of these movement detection algorithms could be improved by devices cooperating during context sensing. Context awareness could also help to classify spatio-temporal clusters for more efficient opportunistic message delivery, e.g. if clusters stationary, mobile, or mixed.

Another application for spatio-temporal cluster detection and PSN research is Distributed Online Social Networking (DOSN). The development of online social networks and collaborative tools such as Twitter, Facebook, and Stack Exchange has resulted in an unprecedented level of correspondence between people in the form of digital media and skills sharing. However, the ownership of the media and information regarding media consumption usually lies with third parties. Often users express privacy concerns by not participating in these online networks, forgoing any benefits they may offer. In order to allow ownership of digital content to remain with the users rather than untrusted third parties, clusters of PMWDs may be able to act as social caches for secure distributed data storage [Han et al., 2012b]. Imagine that a user wants to share a picture with their friends. Instead of uploading the picture to a centralised social networking website such as Facebook, the picture is cached on the user's PMWD and shared during opportunistic encounters between authorised participants of the DOSN. A DOSN as just described is not too far removed from the scenario outlined in Section 1.6 at the start of this thesis. Hopefully the first steps are now being taken towards making PSNs and DOSNs—or something like them—a reality.

# Bibliography

[Abdesslem et al., 2011] Abdesslem, F., Henderson, T., and Parris, I. (2011). The locshare reality mining datasets.

[Abdesslem et al., 2007] Abdesslem, F., Ziviani, A., de Amorim, M., and Todorova, P. (2007). Looking around first: Localized potential-based clustering in spontaneous networks. *Communications Letters, IEEE*, 11(8):653–655.

[Achtert et al., 2012] Achtert, E., Goldhofer, S., Kriegel, H., Schubert, E., and Zimek, A. (2012). Evaluation of clusterings – metrics and visual support. *Proceedings of the international conference on data engineering*, pages 1285–1288.

[Agrawal et al., 1998] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105, New York, NY, USA. ACM.

[Allamanis et al., 2012] Allamanis, M., Scellato, S., and Mascolo, C. (2012). Evolution of a location-based online social network: Analysis and models. In *IMC*, Boston, MA.

[Andrews et al., 2012] Andrews, J., Claussen, H., Dohler, M., Rangan, S., and Reed, M. (2012). Femtocells: Past, present, and future. *IEEE Journal on selected areas in communications*, 30(3):497–508.

[Ankerst et al., 1999] Ankerst, M., Breunig, M., Kriegel, H., and Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. pages 49–60. ACM Press.

[Apolloni et al., 2009] Apolloni, A., Kumar, V. S., Marathe, M. V., and Swarup, S. (2009). Computational epidemiology in a connected world. *Computer*, 42(12):83–86.

[Bakht and Kravets, 2010] Bakht, M. and Kravets, R. (2010). SearchLight: asynchronous neighbor discovery using systematic probing. *ACM SIGMOBILE mobile computing and communications review*, 14(4):31–33.

[Balasubramanian et al., 2010a] Balasubramanian, A., Mahajan, R., and Venkataramani, A. (2010a). Augmenting mobile 3G using WiFi. In *Proceedings of the 8th international conference on mobile systems, applications, and services*, pages 209–222, New York, NY, USA. ACM.

[Balasubramanian et al., 2010b] Balasubramanian, A., Neil Levine, B., and Venkataramani, A. (2010b). Replication routing in DTNs: a resource allocation approach. *IEEE/ACM Transactions on Networking*, 18(2):596–609.

[Banerjee et al., 2010] Banerjee, N., Corner, M., and Levine, B. (2010). Design and field experimentation of an energy-efficient architecture for DTN throwboxes. *IEEE/ACM Trans. Netw.*, 18(2):554–567.

[Bedogni et al., 2012] Bedogni, L., Di Felice, M., and Bononi, L. (2012). By train or by car? detecting the user's motion type through smartphone sensors data. In *2012 IFIP Wireless Days*, pages 1–6.

[Bezdek, 1981] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

[Bigwood et al., 2008] Bigwood, G., Rehunathan, D., Bateman, M., Henderson, T., and Bhatti, S. (2008). Exploiting self-reported social networks for routing in ubiquitous computing environments. In *IEEE international conference on wireless and mobile computing, networking and communications*, pages 484–489.

[Bilmes, 2010] Bilmes, J. (2010). Dynamic graphical models. *Signal Processing Magazine, IEEE*, 27(6):29–42.

[Birrane III et al., 2011] Birrane III, E. J., Burleigh, S. C., and Cerf, V. (2011). Defining tolerance: impacts of delay and disruption when managing challenged networks. Technical report, Jet Propulsion Laboratory, National Aeronautics and Space Administration, Pasadena, CA.

[Bohman et al., 2004] Bohman, D., Frank, M., Martini, P., and Scholz, C. (2004). Performance of symmetric neighbor discovery in bluetooth ad hoc networks. In *GI Jahrestagung (1)'04*, pages 138–142.

[Borgia et al., 2011] Borgia, E., Conti, M., and Passarella, A. (2011). Autonomic detection of dynamic social communities in opportunistic networks. In *IFIP MedHocNet*, pages 142–149, Favignana, Italy.

[Bougard et al., 2005] Bougard, B., Catthoor, F., Daly, D. C., Chandrakasan, A., and Dehaene, W. (2005). Energy efficiency of the IEEE 802.15. 4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In *Design, automation and test in Europe*, pages 196–201.

[Brown et al., 2008] Brown, D., Trinidad, K., and Borja, R. (2008). NASA successfully tests first deep space internet.

[Burns et al., 2005] Burns, B., Brock, O., and Levine, B. (2005). MV routing and capacity building in disruption tolerant networks. In *IEEE International conference on computer communications*, pages 398–408.

[Chaintreau et al., 2005] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., and Scott, J. (2005). Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report University of Cambridge, Computer Lab. UCAM-CL-TR-617.

[Chatfield, 2003] Chatfield, C. (2003). *The analysis of time series: An introduction, sixth edition*. Chapman & Hall/CRC texts in statistical science. Taylor & Francis.

[Choi and Shen, 2011] Choi, B. and Shen, X. (2011). Adaptive asynchronous sleep scheduling protocols for delay tolerant networks. *Mobile Computing*, 10(9):1283–1296.

[Choudhury et al., 2003] Choudhury, T., Clarkson, B., Basu, S., and Pentland, A. (2003). Learning communities: Connectivity and dynamics of interacting agents. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2797–2802.

[Cisco, 2012] Cisco (2012). Cisco visual networking index: Forecast and methodology, 2011-2016. Technical report.

[Contini et al., 2011] Contini, D., Crowe, M., Merritt, C., Oliver, R., and Mott, S. (2011). Mobile payments in the united states mapping out the road ahead.

[Cormen et al., 2011] Cormen, T., Leiserson, C., and Rivest, R. (2011). Section 22.5. In *Introduction to Algorithms, Second Edition*, pages 552–557. MIT Press and McGraw-Hill.

[Daly and Haahr, 2007] Daly, E. and Haahr, M. (2007). Social network analysis for routing in disconnected delay-tolerant manets. In *ACM international symposium on mobile ad hoc networking and computing*, pages 32–40, Quebec, Canada. ACM.

[De Domenico et al., 2012] De Domenico, M., Lima, A., and Musolesi, M. (2012). Interdependence and predictability of human mobility and social interactions. In *Proceedings of the nokia mobile data challenge workshop*, Newcastle, United Kingdom.

[Deloitte and GSMA, 2012] Deloitte and GSMA (2012). Sub-saharan africa mobile observatory 2012. Technical report.

[Demetrescu et al., 2005] Demetrescu, C., Finocchi, I., and Italiano, G. (2005). Chapter 36: Dynamic graphs. In *Handbook on data structures and applications*, Computer and information science. CRC.

[Dutta and Culler, 2008] Dutta, P. and Culler, D. (2008). Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM conference on embedded network sensor systems*, pages 71–84.

[Eagle and Pentland, 2005] Eagle, N. and Pentland, A. (2005). The reality reality mining dataset.

[Eagle and (Sandy) Pentland, 2006] Eagle, N. and (Sandy) Pentland, A. (2006). Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268.

[Easley and Kleinberg, 2010] Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning about a highly connected world*. Cambridge University Press, New York, NY, USA.

[Ekman et al., 2008] Ekman, F., Keranen, A., Karvo, J., and Ott, J. (2008). Working day movement model. In *ACM SIGMOBILE workshop on Mobility models*, pages 33–40, Hong Kong, China.

[Ester et al., 1996] Ester, M., Kriegel, H., S, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press.

[Fidge, 1988] Fidge, C. J. (1988). Timestamps in message-passing systems that preserve the partial ordering. *Proceedings of the 11th Australian Computer Science Conference*, 10(1):56–66.

[Fiore et al., 2007] Fiore, M., Harri, J., Filali, F., and Bonnet, C. (2007). Vehicular mobility simulation for VANETs. In *ANSS*, pages 301–309, Washington, DC. IEEE Computer Society.

[Freeman, 1977] Freeman, L. (1977). Set of measures of centrality based on betweenness. *SOCIOMETRY*, 40(1):35–41.

[Gaito et al., 2012] Gaito, S., Quadri, C., Rossi, G., and Zignani, M. (2012). THINPLE - the new online sociality is built on top of NFC-based contacts. In *IFIP Wireless Days*, Dublin, Ireland.

[Grasic et al., 2011] Grasic, S., Davies, E., Lindgren, A., and Doria, A. (2011). The evolution of a DTN routing protocol - PRoPHETv2. In *ACM workshop on challenged networks*, pages 27–30.

[Han et al., 2012a] Han, B., Hui, P., Kumar, V. S. A., Marathe, M. V., Shao, J., and Srinivasan, A. (2012a). Mobile data offloading through opportunistic communications and social participation. *Mobile Computing, IEEE Transactions on*, 11(5):821–834.

[Han et al., 2012b] Han, L., Punceva, M., Nath, B., Muthukrishnan, S., and Iftode, L. (2012b). SocialCDN: caching techniques for distributed social networks. IEEE.

[Henderson et al., 2008] Henderson, T., Kotz, D., and Abyzov, I. (2008). The changing usage of a mature campus-wide wireless network. *Computer Networks*, 52(14):2690–2712.

[Herbiet and Bouvry, 2010] Herbiet, G. and Bouvry, P. (2010). SHARC: community-based partitioning for mobile ad hoc networks using neighborhood similarity. In *IEEE international symposium on a world of wireless mobile and multimedia networks*, pages 1–9.

[Hooke, 2010] Hooke, A. (2010). Internet predictions. *Internet Computing, IEEE*, 14(1):37–39.

[Huang et al., 2005] Huang, J. H., Amjad, S., and Mishra, S. (2005). CenWits: a sensor-based loosely coupled search and rescue system using witnesses. In *Proceedings of the 3rd international conference on embedded networked sensor systems*, pages 180–191.

[Hui, 2008] Hui, P. (2008). People are the network: experimental design and evaluation of social-based forwarding algorithms. Technical Report University of Cambridge, Computer Lab. UCAM-CL-TR-713.

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

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

[Hui and Crowcroft, 2007] Hui, P. and Crowcroft, J. (2007). How small labels create big improvements. In *IEEE pervasive computing and communications workshop*, pages 65–70.

[Hui and Crowcroft, 2008] Hui, P. and Crowcroft, J. (2008). Human mobility models and opportunistic communications system design. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1872):2005.

[Hui et al., 2007a] Hui, P., Crowcroft, J., and Yoneki, E. (2007a). BUBBLE rap: Social-based forwarding in delay tolerant networks. *Mobile Computing*, 6(1):1576–1589.

[Hui et al., 2007b] Hui, P., Yoneki, E., Chan, S. Y., and Crowcroft, J. (2007b). Distributed community detection in delay tolerant networks. In *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, Kyoto, Japan.

[Hull et al., 2006] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S. (2006). CarTel: a distributed mobile sensor computing system. In *4th ACM SenSys*, Boulder, CO.

[Ingelrest et al., 2007] Ingelrest, F., Mitton, N., and Simplot-Ryl, D. (2007). A turnover based adaptive HELLO protocol for mobile ad hoc and sensor networks. In *International symposium on modeling, analysis, and simulation of computer and telecommunication systems*, pages 9–14, Washington, DC, USA. IEEE Computer Society.

[It, 1993] It, U. (1993). ITU-T recommendation g.114. Technical report, International Telecommunication Union.

[Izumikawa et al., 2010] Izumikawa, H., Pitkanen, M., Ott, J., Timm-Giel, A., and Bormann, C. (2010). Energy-efficient adaptive interface activation for Delay/Disruption tolerant networks. In *ICACT*, volume 1, pages 645–650.

[Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

[Jacquet et al., 2001] Jacquet, P., Mühlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., and Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *IEEE international multi topic conference. technology for the 21st century*, pages 62–68.

[Juang et al., 2002] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, ASPLOS-X, pages 96–107, New York, NY, USA. ACM.

[Kandhalu et al., 2010] Kandhalu, A., Lakshmanan, K., and Rajkumar, R. (2010). U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 350–361, New York, NY, USA. ACM.

[Kelso, 1988] Kelso, T. (1988). *Temporal Clustering in the Multi-Target Tracking Environment*. PhD thesis, The University of Texas, Austin.

[Keranen et al., 2010] Keranen, A., Karkkainen, T., and Ott, J. (2010). Simulating mobility and DTNs with the ONE (invited paper). *Journal of Communications*, 5(2).

[Keranen et al., 2009] Keranen, A., Ott, J., and Karkkainen, T. (2009). The ONE simulator for DTN protocol evaluation. In *International Conference on Simulation Tools and Techniques*, Rome, Italy. ICST.

[Kerry, 2007] Kerry, S. (2007). IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages 1–1233.

[Krishnan and Starobinski, 2006] Krishnan, R. and Starobinski, D. (2006). Efficient clustering algorithms for self-organizing wireless sensor networks. *Ad Hoc Netw.*, 4(1):36–59.

[Lambiotte et al., 2008] Lambiotte, R., Blondel, V., Kerchove, C., Huens, E., Prieur, C., Smoreda, Z., and Dooren, P. (2008). Geographical dispersal of mobile communication networks. *Physica A: Statistical mechanics and its applications*, 387(21):5317–5325.

[LeBrun et al., 2005] LeBrun, J., Chuah, C., Ghosal, D., and Zhang, M. (2005). Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Vehicular Technology Conference 2005*, volume 4, pages 2289–2293.

[Leguay et al., 2007] Leguay, J., Friedman, T., and Conan, V. (2007). Evaluating MobySpace-based routing strategies in delay-tolerant networks. *Wireless Communications and Mobile Computing*, 7(10):1171–1182.

[Leguay et al., 2006] Leguay, J., Lindgren, A., Scott, J., Friedman, T., and Crowcroft, J. (2006). Opportunistic content distribution in an urban setting. In *SIGCOMM workshop on challenged networks*, pages 205–212.

[Leskovec et al., 2005] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. In *The eleventh international conference on Knowledge discovery in data mining*, pages 177–187. ACM.

[Leung et al., 2011] Leung, I. X., Chan, S. Y., Hui, P., and Liò, P. (2011). Intra-city urban network and traffic flow analysis from GPS mobility traces. *arXiv:1105.5839*.

[Leung et al., 2009] Leung, I. X., Hui, P., Liò, P., and Crowcroft, J. (2009). Towards real-time community detection in large networks. *Physical Review E*, 79(6).

[Lin and Wang, 2010] Lin, J. W. and Wang, W. S. (2010). An efficient reconstruction approach for improving bluetree scatternet formation in personal area networks. *Network and Computer Applications*, 33(2):141–155.

[Lindeberg et al., 2012] Lindeberg, M., Haavet, J., Barros, S., Goebel, V., and Plagemann, T. (2012). Message lost or message taken - on message ferry selection in DTNs. In *IFIP Wireless Days*, pages 1–7.

[Lindgren and Hui, 2009] Lindgren, A. and Hui, P. (2009). The quest for a killer app for opportunistic and delay tolerant networks. In *ACM workshop on challenged networks*, pages 59–66.

[Liu and Wu, 2007] Liu, C. and Wu, J. (2007). Scalable routing in delay tolerant networks. In *ACM international symposium on mobile ad hoc networking and computing*, pages 51–60, New York, NY, USA. ACM.

[Liu et al., 2004] Liu, T., Sadler, C. M., Zhang, P., and Martonosi, M. (2004). Implementing software on resource-constrained mobile sensors: Experiences with impala and ZebraNet. In *Proceedings of the 2nd international conference on mobile systems, applications, and services*, pages 256–269.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on information theory*, 28(2):129–137.

[Lu et al., 2003] Lu, C., Chen, D., and Kou, Y. (2003). Detecting spatial outliers with multiple attributes. In *15th IEEE international conference on tools with artificial intelligence*, pages 122–128.

[López-Pérez et al., 2009] López-Pérez, D., Valcarce, A., De La Roche, G., and Zhang, J. (2009). OFDMA femtocells: A roadmap on interference avoidance. *IEEE Communications Magazine*, 47(9):41–48.

[Ma et al., 2008] Ma, Y., Richards, M., Ghanem, M., Guo, Y., and Hassard, J. (2008). Air pollution monitoring and mining based on sensor grid in london. *Sensors*, 8(6):3601–3623.

[MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of MultiVariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

[Madan et al., 2010] Madan, A., Cebrian, M., Lazer, D., and Pentland, A. (2010). Social sensing for epidemiological behavior change. In *ACM conference on ubiquitous computing*, pages 291–300.

[Mansfield and Wright, 2013] Mansfield, G. and Wright, G. (2013). Small cell forum release one: Home. Technical Report 101.01.01, Small cell forum.

[Matsuda and Takine, 2008] Matsuda, T. and Takine, T. (2008). (p,q)-epidemic routing for sparsely populated mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5).

[Milanesi et al., 2013] Milanesi, C., Tay, L., Cozza, R., Atwal, R., Nguyen, T., Tsai, T., Zimmermann, A., and Lu, C. (2013). Forecast: Devices by operating system and user type, worldwide, 2010-2017, 1Q13 update. Technical report, Gartner.

[Milgram, 1977] Milgram, S. (1977). *The Familiar Stranger: An Aspect of Urban Anonymity*. Mcgraw-Hill Book Company, 1st edition.

[Motani and Srinivasan, 2005] Motani, M. and Srinivasan, V. (2005). Peoplenet: engineering a wireless virtual social network. In *Proceedings of ACM MobiCom*, pages 243–257.

[Musolesi and Mascolo, 2009] Musolesi, M. and Mascolo, C. (2009). CAR: context-aware adaptive routing for delay tolerant mobile networks. *IEEE Transactions on Mobile Computing*, 8(2):246–260.

[Natarajan et al., 2007] Natarajan, A., Motani, M., and Srinivasan, V. (2007). Understanding urban interactions from bluetooth phone contact traces. *Passive and Active Network Measurement*, pages 115–124.

[Neill, 2006] Neill, D. (2006). *Detection of spatial and spatio-temporal clusters*. PhD thesis, Carnegie Mellon University.

[Newman, 2004] Newman, M. E. (2004). Analysis of weighted networks. *Physical Review E*, 70(5):056131.

[Nguyen et al., 2011] Nguyen, N., Dinh, T., Tokala, S., and Thai, M. (2011). Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of ACM MobiCom*, pages 85–96, New York, NY, USA. ACM.

[Orlinski and Filer, 2012a] Orlinski, M. and Filer, N. (2012a). Distributed expectation-based spatio-temporal cluster detection for pocket switched networks. In *IFIP Wireless Days*, Dublin, Ireland.

[Orlinski and Filer, 2012b] Orlinski, M. and Filer, N. (2012b). Movement speed based inter-probe times for neighbour discovery in mobile ad-hoc networks. In *Ad Hoc Networks*, volume 111, Paris, France. Springer.

[Orlinski and Filer, 2012c] Orlinski, M. and Filer, N. (2012c). Quality distributed community formation for data delivery in pocket switched networks. In *Annual workshop on simplifying complex networks for practitioners*, pages 31–36, Lyon, France. ACM.

[Orlinski and Filer, 2013] Orlinski, M. and Filer, N. (2013). The rise and fall of spatio-temporal clusters in mobile ad hoc networks. *Ad Hoc Networks*, 11(5):1641–1654.

[Ouchi and Doi, 2012] Ouchi, K. and Doi, M. (2012). Indoor-outdoor activity recognition by a smartphone. In *ACM conference on ubiquitous computing*, pages 600–601, New York, NY, USA. ACM.

[Pan and Saramäki, 2011] Pan, R. K. and Saramäki, J. (2011). Path lengths, correlations, and centrality in temporal networks. 84(1).

[Panisson et al., 2011] Panisson, A., Barrat, A., Cattuto, C., Broeck, W., Ruffo, G., and Schifanella, R. (2011). On the dynamics of human proximity for data diffusion in ad-hoc networks. *Ad Hoc Networks*, 10(8):1532–1543. special issue on social-based routing in mobile and delay-tolerant networks.

[Passarella and Conti, 2011] Passarella, A. and Conti, M. (2011). Characterising aggregate inter-contact times in heterogeneous opportunistic networks. *NETWORKING 2011*, pages 301–313.

[Perkins and Royer, 1999] Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *IEEE workshop on mobile computing systems and applications*, pages 90–100.

[Perrucci et al., 2011] Perrucci, G. P., Fitzek, F. H. P., and Widmer, J. (2011). Survey on energy consumption entities on the smartphone platform. In *73rd vehicular technology conference*, pages 1–6. IEEE.

[Petrioli et al., 2004] Petrioli, C., Basagni, S., and Chlamtac, I. (2004). BlueMesh: degree-constrained multi-hop scatternet formation for bluetooth networks. *Mobile Networks and Applications*, 9(1):33–47.

[Pietilainen and Diot, 2012] Pietilainen, A. and Diot, C. (2012). Dissemination in opportunistic social networks: the role of temporal communities. In *ACM international symposium on mobile ad hoc networking and computing*, pages 165–174, South Carolina, USA.

[Pietilainen et al., 2008] Pietilainen, A. K., Oliver, E., Lebrun, J., Varghese, G., Crowcroft, J., and Diot, C. (2008). Experiments in mobile social networking. Technical report.

[Reddy et al., 2011] Reddy, S., Estrin, D., and Srivastava, M. (2011). Network services for mobile participatory sensing. *Emerging wireless technologies and the future mobile Internet*, page 154.

[Ross, 2012] Ross, P. (2012). Phone-y money. *Spectrum, IEEE*, 49(6):60 –63.

[Royer and Toh, 1999] Royer, E. M. and Toh, C. K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE personal communications*, 6(2):46–55.

[Scellato et al., 2011a] Scellato, S., Leontiadis, I., Mascolo, C., Basu, P., and Zafer, M. (2011a). Understanding robustness of mobile networks through temporal network measures. In *IEEE International Conference on Computer Communications*, pages 1–5.

[Scellato et al., 2011b] Scellato, S., Musolesi, M., Mascolo, C., Latora, V., and Campbell, A. (2011b). Nextplace: A spatio-temporal prediction framework for pervasive systems. *Pervasive computing*, pages 152–169.

[Schikuta and Erhart, 1997] Schikuta, E. and Erhart, M. (1997). The BANG-clustering system: Grid-based data analysis. In *Proceedings of the second international symposium on advances in intelligent data analysis, reasoning about data*, pages 513–524. Springer-Verlag.

[Scott et al., 2009] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. (2009). The haggle reality mining datasets.

[Scott, 2007] Scott, K. (2007). RFC 5050 - bundle protocol specification.

[Shah et al., 2003] Shah, R., Roy, S., Jain, S., and Brunette, W. (2003). Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2-3):215–233.

[Sharma et al., 2007] Sharma, G., Mazumdar, R., and Shroff, B. (2007). Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. *Trans. Netw.*, 15(5):981–992.

[SIG, 2010] SIG, B. (2010). Bluetooth specification version 4.0 - master table of contents & compliance requirements. Technical report.

[Small and Haas, 2005] Small, T. and Haas, Z. (2005). Resource and performance tradeoffs in delay-tolerant wireless networks. In *SIGCOMM*, pages 260–267.

[Sokal, 1963] Sokal, R. (1963). The principles and practice of numerical taxonomy. *Taxon*, 12(5):190–199.

[Spyropoulos et al., 2004] Spyropoulos, T., Psounis, K., and Raghavendra, C. (2004). Single-copy routing in intermittently connected mobile networks. In *First Annual IEEE Communications Society Conference on sensor and ad hoc communications and networks*, pages 235–244.

[Spyropoulos et al., 2005] Spyropoulos, T., Psounis, K., and Raghavendra, C. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant networking*, pages 252–259, New York, NY, USA. ACM.

[Spyropoulos et al., 2008] Spyropoulos, T., Psounis, K., and Raghavendra, C. (2008). Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Transactions on networking*, 16(1):77–90.

[Sulo et al., 2010] Sulo, R., Berger-Wolf, T., and Grossman, R. (2010). Meaningful selection of temporal resolution for dynamic networks. In *Workshop on mining and learning with graphs*, pages 127–136, Washington, D.C.

[Trauwaert, 1988] Trauwaert, E. (1988). On the meaning of dunn's partition coefficient for fuzzy clusters. *Fuzzy Sets and Systems*, 25(2):217–242.

[Troël, 2004] Troël, A. (2004). *Prise en compte de la mobilité dans les interactions de proximité entre terminaux à profils hétérogènes*. PhD thesis, Université de Rennes I.

[van Eenennaam et al., 2012] van Eenennaam, M., van de Venis, A., and Karagiannis, G. (2012). Impact of IEEE 1609.4 channel switching on the IEEE 802.11p beaconing performance. In *IFIP Wireless Days*, pages 1–8.

[Vasudevan et al., 2005] Vasudevan, S., Kurose, J., and Towsley, D. (2005). On neighbor discovery in wireless networks with directional antennas. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 4, pages 2502–2512.

[Vu et al., 2010] Vu, L., Nahrstedt, K., Retika, S., and Gupta, I. (2010). Joint bluetooth/wifi scanning framework for characterizing and leveraging people movement in university campus. In *ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, pages 257–265.

[Wang et al., 2009a] Wang, W., Motani, M., and Srinivasan, V. (2009a). Opportunistic energy-efficient contact probing in delay-tolerant applications. *Trans. Netw.*, 17(5):1592–1605.

[Wang et al., 2007] Wang, W., Srinivasan, V., and Motani, M. (2007). Adaptive contact probing mechanisms for delay tolerant applications. In *Proceedings of ACM MobiCom*, pages 230–241, New York, NY, USA. ACM.

[Wang et al., 1997] Wang, W., Yang, J., and Muntz, R. (1997). *STING: A Statistical Information Grid Approach to Spatial Data Mining*.

[Wang et al., 2009b] Wang, Y., Krishnamachari, B., and Valente, T. (2009b). Findings from an empirical study of fine-grained human social contacts. In *WONS*, pages 153–160.

[Whitbeck and Conan, 2010] Whitbeck, J. and Conan, V. (2010). HYMAD: hybrid DTN-MANET routing for dense and highly dynamic wireless networks. *Computer Communications*, 33(13):1483–1492.

[Williams et al., 2012] Williams, M. J., Whitaker, R. M., and Allen, S. M. (2012). Decentralised detection of periodic encounter communities in opportunistic networks. *Ad Hoc Networks*, 10(8):1544–1556. special issue on social-based routing in mobile and delay-tolerant networks.

[Wirtz et al., 2012] Wirtz, H., Martin, D., Grap, B., and Wehrle, K. (2012). On-demand content-centric wireless networking. In *Proceedings of ACM MobiCom*, pages 451–454, New York, NY, USA. ACM.

[Yang et al., 2009] Yang, D., Shin, J., Kim, J., and Kim, C. (2009). Asynchronous probing scheme for the optimal energy-efficient neighbor discovery in opportunistic networking. In *IEEE pervasive computing and communications workshop*, Washington, DC.

[Ye et al., 2002] Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Twenty first annual joint conference of the IEEE computer and communications societies.*, volume 3, pages 1567–1576.

[Yoon et al., 2003] Yoon, J., Liu, M., and Noble, B. (2003). Random waypoint considered harmful. In *IEEE International Conference on Computer Communications*, volume 2, pages 1312–1321.

[Yu and Cai, 2010] Yu, F. and Cai, R. (2010). Optimized fuzzy information granulation of temporal data. In *Seventh international conference on fuzzy systems and knowledge discovery*, pages 419–423.

[Zare et al., 2013] Zare, A., Taheri, H., and Moghaddam, M. (2013). Using limited flooding in on-demand distance vector junior for reduction power consumption in ZigBee networks. In Das, V. and Chaba, Y., editors, *Mobile communication and power engineering*, volume 296 of *Communications in computer and information ccience*, pages 103–108. Springer Berlin Heidelberg.

[Zhang et al., 2006] Zhang, X., Neglia, G., Kurose, J., and Towsley, D. (2006). Performance modeling of epidemic routing. *Comput. Netw.*, 51(10):827–839.

[Zhao et al., 2004] Zhao, W., Ammar, M., and Zegura, E. (2004). A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198, New York, NY, USA. ACM.

[Zhou et al., 2007] Zhou, D., Councill, I., Zha, H., and Lee Giles, C. (2007). Discovering temporal communities from social network documents.

[Zyba et al., 2011] Zyba, G., Voelker, G., Ioannidis, S., and Diot, C. (2011). Dissemination in opportunistic mobile ad-hoc networks: The power of the crowd. In *IEEE international conference on computer communications*, pages 1179–1187.