Content Dissemination in Participatory Delay Tolerant Networks

Afra Jahanbakhsh Mashhadi

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University College London.

Department of Computer Science

University College London

2011

I, Afra Jahanbakhsh Mashhadi confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

As experience with the Web 2.0 has demonstrated, users have evolved from being only consumers of digital content to producers. Powerful handheld devices have further pushed this trend, enabling users to consume rich media (for example, through high resolution displays), as well as create it on the go by means of peripherals such as built-in cameras.

As a result, there is an *enormous* amount of user-generated content, most of which is relevant only within local communities. For example, students advertising events taking place around campus. For such scenarios, where producers and consumers of content belong to the same local community, networks spontaneously formed on top of colocated user devices can offer a valid platform for sharing and *disseminating* content.

Recently, there has been much research in the field of content dissemination in mobile networks, most of which exploits user mobility prediction in order to deliver messages from the producer to the consumer, via spontaneously formed Delay Tolerant Networks (DTNs). Common to most protocols is the assumption that users are willing to participate in the content distribution network; however, because of the energy restrictions of handheld devices, users' participation cannot be taken for granted.

In this thesis, we design content dissemination protocols that leverage information about user mobility, as well as interest, in order to deliver content, while avoiding overwhelming non-interested users. We explicitly reason about battery consumption of mobile devices to model participation, and achieve fairness in terms of workload distribution. We introduce a dynamic priority scheduling framework, which enables the network to allocate the scarce energy resources available to support the delivery of the most desired messages. We evaluate this work extensively by means of simulation on a variety of real mobility traces and social networks, and draw a comparative evaluation with the major related works in the field.

Acknowledgements

Pursuing a PhD. is a considerable undertaking, one that cannot be achieved alone, and there are many people that I would like to thank for their feedback and support during this time. I have been fortunate to be given the opportunity to pursue this path in the department of Computer Science at University College London, alongside many bright students with whom I have shared great moments, and staff who have provided me with significant assistance over the years.

In particular, I am forever grateful for the guidance and encouragement of my supervisor, Licia Capra, whose endless patience is beyond description. This work would not have been possible without her insights and knowledge. I am also grateful to Stephen Hailes and David Rosenblum for their constructive feedback, and my colleague and dear friend, Sonia Ben Mokhtar, for the valuable support she gave me when I most needed it.

I would also like to thank Franco Raimondi, Daniele Quercia, Neal Lathia, Liam McNamara, Anders Lindgren, Christian Wallenta and Panteha Saeedi for their technical help and interesting discussions on various topics related to my thesis. My profound gratitude also goes to Clovis Chapman, for whose constructive comments and support I am deeply indebted. Finally, I would also like to thank the Deutsch Telekom Labs and in particular Pan Hui for providing me an opportunity to have a taste of research in industry.

This thesis is dedicated to my parents, whose love and unconditional support has always been tremendous; and to my grandfather who by now knows quite a bit about delay tolerant networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, two phenomena of massive proportions have emerged: first, the transformation of the Internet users from passive consumers to active producers of digital content (e.g., personal stories, videos, pictures, etc.), as witnessed by the popularity of Web 2.0 websites. Second, and almost in parallel, mobile technology has undergone a major evolution. Portable devices (e.g., smart phones, portable digital assistants, etc.) have seen their computing capabilities such as processing power and memory availability, grow according to Moore's law, while also accommodating a variety of wireless network interfaces of increasing bandwidth (e.g., Wi-Fi and Bluetooth 2). Moreover, additional peripherals such as built-in digital cameras, have become commodity on such devices.

Convergence of these two trends, in conjunction with characteristics of urban lifestyle (e.g., where people spend considerable amount of their time in public spaces or commuting), has already made mobile devices a comfortable platform for consuming content, such as videos, music, and pictures on the go. Indeed, researchers have forecasted that mobile data traffic will grow at an annual growth rate of 108 percent between 2009 and 2014 [Cisco Visual Networking Index, 2010], predicting video traffic itself to be responsible for 66% of the growth. This is not surprising, considering the popularity of websites such as YouTube.com, and the fact that users are now able to produce high quality video clips anywhere and anytime, using their mobile phones, and view them on the go by means of 3G and Wi-Fi Internet connectivity.

For such user-generated content, there exist scenarios where the produced and consumed content both belong to the same local community. For example, in a campus-based scenario, students looking for flatmates may advertise their requests, together with pictures of the flat; student bands may disseminate samples of their music, together with information about upcoming gigs; events organised by the various student societies can be advertised by means of electronic flyers. Indeed, in a metropolitan city like London, an estimated average of 5,000 social events take place every day; however, only (less than) half of these are being listed on popular websites (thus accessible via 3G networks from users mobile phones). The remainders are still being advertised by word-of-mouth, posters affixed in given areas, hand-distributed flyers, and the like. For those unlisted user-generated advertisements (content) to be placed on the World Wide Web, there exists an associated start-up cost which is directly imposed on the user (or the community) despite the availability of cheap hosting services. This cost corresponds to the initial construction cost of any business, offering a particular service, along with the time and effort that is required for making people aware of the offered services and getting magnitude of users to sign up to the website.

Even assuming this start-up cost could be afforded by any business, *consuming* digital content on the go by means of 3G services is still fairly limited and expensive. In recent years, although 3G services have become more available, their coverage and quality has remained an ongoing hurdle. Furthermore, as these services become more widely used (thanks to the widespread market penetration of compatible smartphones), they do not seem to be able to cope with the corresponding high demand [Wortham, 2009]. Indeed, to address this increased demand, network operators have all enforced limits on the amount of data traffic that each user can consume [Ziegler, 2010].

To facilitate the sharing of user-generated content on the fly, without imposing unaffordable cost and/or limitation to users, a dynamic and distributed content sharing platform is desired. For such scenarios, mobile networks formed by mobile devices during periods of colocation can offer a more efficient and effective way of disseminating content amongst the local community. By more efficient, we mean that content could be distributed at *zero cost* both in terms of human resources and actual financial cost, by exploiting the short range connections (such as Wi-Fi and Bluetooth 2) available on people smartphones. By more effective, we mean that content that could previously be propagated only by traditional ways, such as word-of-mouth, can now be shared, and thus a larger pool of *interested* people could be notified of a relevant event.

In the past decade, a new form of communication called Delay Tolerant Networks (DTNs) has emerged. Delay tolerant networks are, in essence, distributed networks formed over a multitude of devices that are dynamically connected, without any fixed or stable connection. Initially, applications of delay tolerant networks were concentrated on challenged environments, such as emergency scenarios or underwater zones. Indeed, the very first application of DTNs was proposed for interplanetary exploration [Burleigh et al., 2003]. It has not been until very recently that the view of the target environment has been broadened, to incorporate other possible fields in which delay tolerant networks would be highly desirable. Today, DTN applications have been extended to incorporate information dissemination in human networks (i.e., Pocket Switch Networks [Hui et al., 2005], spontaneously formed on top of mobile devices embedded in user's daily life), wild life monitoring [Mainwaring et al., 2002, Lindgren et al., 2008], and so on. In the context of human networks, until very recently, the suitability of DTN applications was limited to small data transfers. This was due to the limitations imposed by Bluetooth technology, the only available point-to-point communication interface between portable devices [Shorey and Miller, 2002]. Bluetooth's limited bandwidth and connectivity range creates a bottleneck on data transfer, especially if periods of colocation are short, as it is often the case in delay tolerant networks. However, the recent launch of Wi-Fi direct [Wi-Fi Alliance, 2010] has lifted this restriction, allowing rich media content to be transferred, by means of higher bandwidth and longer connectivity range, thus making delay tolerant networks an even more attractive platform for the spontaneous sharing of media content among local communities.

Many routing protocols have been proposed to facilitate content dissemination in DTN. The very first generation of these protocols focused on achieving high delivery. In this regard, protocols based on pure flooding such as Epidemic [Vahdat and Becker, 2000] were introduced. However, they proved to cause significant overhead, especially when the shared content is of media type and bulky in size. Consequently, the research community has looked into probabilistic routing, that is, a sort of controlled flooding, as a way to limit overhead; however these protocols performed well when evaluated on random mobility models. As the real traces of human mobility started to become available, various studies highlighted fundamental differences between the random and real mobility traces, causing the performance of the probabilistic routing protocols to be negatively impacted when using the real mobility traces. Recently, research studying human mobility patterns suggested that indeed human movement has a high degree of predictability [Rhee et al., 2008, Song et al., 2010]. As a result, the DTN community has turned its direction towards exploiting users' regularity of movement, in order to deliver content effectively and efficiently [Chaintreau et al., 2007, Costa et al., 2008]. Common to all these works is the assumption that users are always willing to participate, and relay the given content. However, this is a big assumption which is unlikely to hold. This is because mobile phones are energy-constrained devices which would suffer greatly, should they be asked to forward many messages (i.e., their batteries would be depleted very quickly). If such assumption is removed, the delivery of state-of-the art protocols would thus be highly impacted. This opens up a new and challenging research problem, which we discuss in more details next.

1.1 Research Problem

To better describe the research problem that we tackle in this thesis, let us consider a large university campus scenario, where a rich variety of events are organised throughout the year by various clubs and societies (University College London has almost 200 clubs and societies [UCLU, 2010]). Such events can be advertised by means of multimedia flyers, and disseminated via DTNs formed over staff and students' mobile phones to the local community. For this to happen, content often needs to traverse a multi-hop path, through the mobile devices of other users in the network, so to compensate for the lack of direct encounter between the producer and the consumer of content.

From research on the Web 2.0 [Cha et al., 2007], we know that there exists a power law distribution of interests, with a small portion of content being of interest to many users, and the vast majority of it being of interest to only small groups of users instead. For this latter category, in order to reach interested people, content would thus have to traverse intermediaries, who may not be interested in the content at all, and who are asked to forward it solely due to their position in the network (i.e., being in between the source and the destination of the message). Considering the bulky nature of the shared content (i.e., video clips in the order of megabytes), routing such amount of non-relevant content will have a direct impact on devices' batteries. This impact cannot be neglected, as it can potentially cause users to stop participating in the content delivery network altogether, causing the user-driven DTN network to collapse.

This problem is unlikely to be solved by advances in the available technology; indeed, [Paradiso and Starner, 2005] have shown that battery resources have reached their maximum capacities, and Moore's Law growth will apply to the miniaturisation of battery size, rather than to an increase of their lifetime.

Hence, for DTN to be a viable means for distributing content in human networks, it is crucial to overcome this problem by accounting for users' participation in the network. We next list the challenges associated with tackling this problem, which we label for future references.

Recipients (Ch 1). First of all, we need to know who is interested in receiving what content. This information is often available as a user's profile. In centralised settings (e.g., such as Twitter and Facebook), this information is always accessible, but in DTN where there is no central server to rely on, each device holds its own user's profile. Thus, we need to find a way of sharing this information in a local community, in order to know *where* content should flow.

Routing (Ch 2). Second, once we know the interest network, our second challenge is to find a (multi-hop) path to deliver content to interested recipients, so that there is a high probability of delivering content, while at the same time seeking participation from *interested intermediaries* only (as uninterested intermediaries have a higher probability of not being willing to contribute as they receive no benefit from it).

Load Distribution (Ch 3). Finding paths that cross interested intermediaries is necessary, but not always sufficient, especially in scenarios where a large amount of content has to be relayed using resource constrained devices. In such cases a node's interest in receiving content does not imply his willingness to forward it. However, in urban areas, the chances are that there exists multiple paths to bring content to destinations. We thus must make sure the load is distributed fairly among these paths, and direct traffic towards the least loaded intermediaries as a further way to ensure participation.

Resource Allocation (Ch 4). Despite taking load-aware decisions on what paths content should follow, battery resources are limited, and can only cover for a subset of the required data transfers. A fourth and final challenge arises as to how best use the few resources available, so to deliver those messages that end-users value the most, thus increasing overall network satisfaction.

1.2 Research Hypothesis and Objectives

Our overall research hypothesis states that:

"By reasoning explicitly on users' interest (application information) and device characteristics (physical information), we can build a source-based content distribution protocol that achieves a high effectiveness and efficiency, while also lifting the unrealistic assumption of unlimited resources and unquestionable users' participation."

More precisely:

- by reasoning on users' *interests* and *mobility patterns*, we can build a *source-based* routing protocol that limits the use of uninterested intermediaries (efficient) while also keeping delivery (effectiveness) as high as state-of-the-art protocols;
- by reasoning on available *battery*, we can build a load-balancing technique that ensures each node in the network carries out its share of workload, thus promoting fairness;
- by reasoning on users' *interests* and *mobility patterns*, we can prioritise messages and build a protocol that allocates available resources to the delivery of the most valued messages, thus achieving high satisfaction.

Therefore, our main objective is to achieve effectiveness, efficiency, fairness, and satisfaction within the context of content dissemination networks.

Effectiveness refers to the number of messages that the source-based protocol manages to deliver to the interested recipients in the network.

Efficiency refers to the reduction of overhead, where overhead is measured not only as the number of replica messages active in the network, but also as the number of messages that the uninterested intermediaries have to relay.

Fairness corresponds to the distribution of demand on participants to route messages, and is measured as workload variance in the network.

Satisfaction corresponds to the total value of messages that were delivered by the routing protocol from an end-user perspective.

We verify our hypothesis by means of simulation on various real networks with different mobility topologies and distributions of user interests. We validate our hypothesis by performing a thorough comparative evaluation of the proposed source-based protocols against the best known protocols in the field.

1.3 Scope and Assumption

Throughout this thesis, when we refer to *"scarce resources"* on mobile devices we specifically refer to energy constraints, while we assume storage suffices at all times. We regard storage management to be outside the scope of this thesis as research has shown that Moore's Law applies to storage capacities; indeed, modern Android phones and iPhones already have up to 30GB storage. As previously mentioned, battery constraints on mobile phones are not expected to follow the same rate of growth instead, as Moore's Law will apply to the miniaturisation of battery size rather than increasing its lifetime [Paradiso and Starner, 2005]. Hence, energy efficiency can be considered currently the most important resource limitation for participatory networks.

We further assume that, when colocation occurs, it lasts long enough for data transfers to take place, assuming sufficient bandwidth for the messages to be transferred over the medium. For instance, assuming Wi-Fi Direct Bandwidth [Wi-Fi Alliance, 2010] of 24MBps, it would take over 4 seconds to transfer a piece of content 100MB in size. However, in practice, due to interference, the transmission often takes longer than estimated, but is still short enough. Furthermore, should the *duration of the colocation* matter, this work could be integrated with research focused on bandwidth sharing in content dissemination networks [McNamara, 2009].

Our focus throughout this work is to cater for users' participation; as such, we directly deal with users' selfishness, that is, those users not participating in order to save battery. However, we do not dwell into malicious behaviour that users might follow to disrupt the network (e.g., relaying messages through a wrong path, or dropping messages).

We are aware that this work touches upon some *privacy* issues, such as gathering and sharing information about how people move, their likes and dislikes, etc. Although there exists an increasing number of people for whom privacy is indeed a less sensitive issue [Pogue, 2010], to promote acceptance of the content dissemination networks, it is important to account for protecting user's privacy. In this regard, there exists many works in protecting privacy in mobile networks, especially concerning location sharing [Gedik and Liu, 2007, Gedik and Liu, 2005, Consolvo et al., 2005]. However, we do not touch upon this issue as it deserves separate and thorough research.

Throughout this thesis we use various keywords in order to refer to participants in the network. We refer to these participants as *"nodes"* when accounting for their functional behaviour in a distributed setting (e.g., in the context of routing); we use the keyword *"user"* when referring to human behaviour of the participants, such as their interest or selfishness; and finally when referring to participants' resources and hardware constraints we use the term *"devices"*. We assume each user is associated with only one device, hence identifiable in the network by it. We also refer to shared content as a message, and assume that the content can fit into one message without need of fragmentation. The terms *"load"* and *"workload"* are used interchangeably, describing the forwarding action required by a node, as we will detail in Chapter 5.

1.4 Thesis Contributions

This thesis contributes to the field of Delay Tolerant Networks by introducing a new stream of research in which user participation is taken into account. Within this research stream, we make the following main contributions:

- A source-based routing protocol that combines mobility information and users' interests, in order to achieve high delivery while not overwhelming uninterested nodes.

- An adaptive load-balancing mechanism that, once integrated with source-based routing protocols, achieves fair workload distribution over time, without compromising delivery.

- A dynamic scheduling mechanism that combines opportunistic reasoning and message importance, in order to increase overall network-wide user satisfaction, without compromising delivery.

- An extensive evaluation by means of simulation on various real datasets of both social networks and human mobility traces, as well as a comparative evaluation with major protocols in the field.

1.5 Thesis Structure

The rest of this thesis is structured as follows:

Chapter 2 reports on the state-of-the-art of content dissemination research. It first examines three main fields, namely peer-to-peer, publish-subscribe, and ad hoc networks; it then dwells into delay tolerant networks, critically challenging their suitability to the environment and the stated research problem.

Chapter 3 presents a unified model for this thesis, as well as detailing assumptions regarding the environment on which the thesis focuses.

Chapter 4 presents the foundation of this work, by proposing an interest-aware content dissemination protocol which takes into account user participation (tackling Ch 1 and Ch 2).

Chapter 5 presents our proposed load-balancing method, which takes devices' energy resources into account and fairly distributes workload among participants (tackling Ch 3).

Chapter 6 introduces a priority scheduling framework which allocates the scarce resources of devices into forwarding the most valued messages first (tackling Ch 4).

Chapter 7 contains our concluding remarks and proposes directions for future work in the field of mobile content dissemination.

Chapter 2

Literature Review

Content dissemination has been a widely researched topic by the networking community. In this chapter, we focus on the state-of-the-art of content dissemination in distributed systems; we group the literature into two categories: *traditional* (or fixed) and *mobile* networks. By traditional networks, we refer to distributed systems in which nodes (often referred to as hosts) do not experience physical movement. In other words, even though nodes may temporarily disconnect from the network, the underlying fixed network (e.g., LAN) persists at all times. On the contrary, nodes in mobile networks act independently from the underlying network, and connections are only formed spontaneously by means of short range device-to-device communication interfaces, such as Bluetooth.

The rest of this chapter is structured as follows: in Section 2.1, we give a brief background on content dissemination in distributed systems over fixed networks. Particularly, we focus on the two most successful paradigms for content sharing, *peer-to-peer* and *publish-subscribe*, explaining the traditional architectures underlying each of these. In Section 2.2, we focus on the related literature specifically introduced to cater for the topological properties of mobile networks. Before reviewing these solutions, we first examine the attempts to port the traditional paradigms such as publish-subscribe to the mobile settings (Section 2.2.1). We then categorise the literature in two: Ad Hoc Networks, describing relatively connected and stable topologies (in particular, we focus on mobile ad hoc networks and vehicular ad hoc networks in Section 2.2.2); and Delay Tolerant Networks (DTNs), where connections are of short duration and much more dynamic. For the latter category, in Section 2.2.3, we review the research undergone so far in the field, describing its suitability to fit human networks, and detailing some of the major routing protocols in this field.

2.1 Content Dissemination in Traditional Networks

In this section, we focus on traditional approaches to content dissemination in distributed systems. Despite the fact that they have been developed for different scenarios, their main concept are applicable also in mobile networks, and indeed as we shall see in Section 2.2, the first content dissemination protocols for mobile networks were extensions of these. In this regard, we describe two successful paradigms: peer-to-peer and publish-subscribe.

Peer-to-Peer. The peer-to-peer paradigm can be seen as a distributed architecture designed for sharing files by directly exchanging them between hosts, rather than requiring the intermediary support of a centralised authority [Androutsellis-Theotokis and Spinellis, 2004]. Although this definition can be applied to many other distributed architectures (such as Grid systems), what distinguishes a peer-to-peer architecture is its ability to deal with instability, transient populations, fault tolerance, and self adaptation. Content distribution is an important application of the peer-to-peer paradigm, which has been proven extremely successful, specifically in the Internet domain [Rodrigues and Druschel, 2010]. The best known and original protocols for content distribution include BitTorrent [Cohen, 2003], PPLive [Hei et al., 2007] and CoolStreaming [Zhang et al., 2005]. Examples of content distribution applications based on the peer-to-peer paradigm include Kazaa[1], Limewire[2], Spotify[3], and RawFlaw[4].

The peer-to-peer architecture assumes that hosts are aware of the specific content they would like to gather. More specifically, users request a specific piece of content (e.g., a music file); the peer to peer architecture is then used in order to locate the requested files among self-configured distributed hosts, thus *pulling* the requested content towards the end-user. As such, it does not suit scenarios in which users are not aware of what is being produced and offered in the network (in other words, scenarios in which information needs to be *pushed* to users proactively).

Publish-Subscribe. In scenarios where content should be *pushed* to users, the publish-subscribe paradigm has proved to be a very successful one [Rosenblum and Wolf, 1997, Carzaniga et al., 2000, Cugola et al., 2002]. More specifically, publish-subscribe networks offer a de-coupled architecture for content dissemination in distributed settings, where the producer and consumer of the content may not know each other; thus allowing

[1] www.kazaa.com
[2] www.limewire.com
[3] www.spotify.com
[4] www.rawflow.com

content to reach a bigger pool of users who are interested in it, without necessarily knowing the producer (as in line with our research objective discussed in Chapter 1). In these networks, users subscribe to content by placing the subscriptions on intermediary nodes, *brokers*, which are the nodes responsible for identifying recipients for the published content, and finding routes to reach them. Brokers subsequently *pull* and gather content and subscriptions from the network, and notify the related subscribers upon a successful content-subscription matching. To do so, brokers maintain routing tables, describing how to reach each subscriber node in the network.

Although the publish-subscribe paradigm offers a de-coupled architecture suitable in principle for any distributed setting (such as mobile networks), the solutions based on this paradigm have mainly focused on traditional networks, where a strong assumption about stability of connections exists and is relied heavily upon, for instance in maintaining routing tables. In the next section, we review recent research, attempting to port this paradigm to a mobile setting.

2.2 Content Dissemination in Mobile Networks

In this section we review the state-of-the-art of content dissemination in mobile networks, by which we specifically refer to *human* mobile networks, formed via their mobile devices and defined by their mobility. We first review the attempts to port publish-subsribe paradigm into these networks (Section 2.2.1), and reason on their suitability in the mobile settings. We then focus on specific solutions designed for mobile networks by reviewing research in the field of Ad Hoc Networks (Section 2.2.2). Finally we examine the state-of-the-art of Delay Tolerant Networks (Section 2.2.3).

2.2.1 Mobile Publish-Subscribe

Mobile networks differ from traditional settings substantially in terms of their topological properties. Human movement causes the connections to be frequently broken, creating a detrimental effect on traditional solutions. For instance, porting publish-subscribe directly to mobile settings creates an excessive overhead associated with maintaining routing tables, due to frequent disconnection.

Recently, there has been various research focusing on how to successfully port the publish-subscribe paradigm to mobile environments [Baldoni et al., 2005, Yoneki et al., 2007a, Ioannidis et al., 2009]. In [Baldoni et al., 2005], the authors propose a fully distributed content-based routing protocol for mobile networks, under the assumption of a highly dynamic environment. Each broker periodically broadcasts a beacon message containing all its subscriptions, thus allowing brokers to associate a *proximity value* to the previously encountered brokers. The routing is then done based on the *proximity* tables. However, the authors assume that nodes are highly dynamic and thus encounter many neighbours at any given time, allowing them to be continuously connected to a subset of nodes in the network. This assumption does not apply in scenarios where mobility is defined by human movement and the nodes suffer from relatively long periods of disconnection and network's sparsity.

Other research aiming to port the publish-subscribe paradigm to mobile networks relies on a semi-distributed architecture. In [Yoneki et al., 2007a], the authors introduce a publish-subscribe approach based on uncovered community structures. In particular, they address scenarios in which brokers are not pre-defined nodes in the network; rather they are elected based on their social characteristic within communities. To do so, communities are first detected, and within each community a broker is chosen. It is further assumed that brokers have permanent stable connections between themselves, thus forming a semi-distributed network. Each broker is then in charge of maintaining subscriptions and routing information for all the other nodes in its own community only. Similarly, [Ioannidis et al., 2009] relies on a semi-distributed architecture where the network operators act as backbones, in order to support the brokers. The brokers are thus chosen by network operators amongst well connected mobile users (i.e., the nodes with the highest in-degree).

Although these works focus on human networks, they rely on a semi-distributed setting assuming the support of a central authority, such as the network operator. In practice, such services involving network operators often impose an unavoidable cost on users. Furthermore, ordinary users who are selected based on their topological or social characteristics to act as brokers, are assumed to be willing to participate, allocating their scarce device resources (e.g., battery), as well as accepting the cost associated with the provided service, to benefit other users in the network.

2.2.2 Ad Hoc Networks

Ad hoc networks, such as Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs) can be defined based on their major property of independency from any fixed infrastructure for communication, with common properties such as self-configuration, self-managment, and short transmission range.

MANETs can be employed in an environment where nodes may move arbitrarily, for example in an office, to allow establishing connectivity among portable devices. According to a survey of MANET [Mauve et al., 2002], routing can be categorised into: *topology-based* and *position-based* routing. Topology-based routing uses information about the links that exist in the network to perform packet forwarding. These routing protocols can behave either proactively (such as DSDV [Perkins and Bhagwat, 1994], OLSR [Clausen and Jacquet, 2003]), by maintaining routing information about the available paths in the network, even if these paths are not currently used, or reactively (DSR [Johnson et al., 2002], AODV [Perkins et al., 2003]), by maintaining only information about routes which are currently in use. The main drawback of MANET routing protocols is that the maintenance of paths can result in a significant amount of traffic, especially in cases where the topology of the network changes frequently, such as observed in human mobility [Das et al., 2000]. In response to this drawback, position-based routing algorithms [Navas and Imielinski, 1997, Ko and Vaidya, 2000, Li et al., 2000] have been introduced, which exploit additional information about the physical position of the participating nodes, in order to find paths to destinations. They rely on the assumption that nodes' positioning information is available thorough the use of a positioning service, thus limiting their suitability to semi-distributed networks only (i.e., requiring a central positioning server).

Vehicular Ad hoc Networks are a special case of MANETs, focusing on inter-vehicle communication, with applications such as improving road traffic and safety [Li and Wang, 2008]. In terms of topological properties such as stability of connections, VANETs exhibit closer similarity to human networks, as (unlike MANETs) they do not assume the existence of long connections amongst nodes. Indeed, VANETs differ from MANETs by their highly dynamic topology, causing MANET routing protocols to have poor convergence and low communication throughput when applied to a vehicular setting. Given that vehicle movement is usually restricted to just bidirectional movement and is constrained to roads and streets, a new stream of geographic routing research has emerged (as opposed to topology-based routings, such as AODV [Perkins et al., 2003] and DSR [Johnson et al., 2002]).

Protocols developed for VANETs have very focused applicability: they are only suitable where mobility is restricted (e.g., bidirectional movement). Moreover, applications are mostly focused on *forwarding* information such as traffic updates to *geographically identified* destinations. This is in contrast to content dissemination in human networks, where the task of discovering who is interested in a piece of content by itself is one of the most important challenges of any routing protocol.

2.2.3 Delay Tolerant Networks

Delay Tolerant Networks (DTNs) are different from Mobile Ad Hoc Networks (MANETs), as they describe settings where connectivity among nodes does not exist, and when it does, it is often for very short periods of time. The past decade has seen significant research in the field of DTNs. In this section, we briefly overview the state-of-the-art of DTN routing protocols, before drawing our focus on a more detailed description of the major DTN routing protocols related to this research. We refer the interested reader to [D'Souza and Jose, 2010] for a full survey on routing approaches in delay tolerant networks.

The first generation of DTN protocols [Broch et al., 1998, Vahdat and Becker, 2000] assumed human movement followed the random waypoint model [Bettstetter et al., 2003], focusing on achieving high delivery in these sparse networks where nodes' mobility significantly contributes to the disconnected nature of the network. In this regard, Vahdat et al. [Vahdat and Becker, 2000] proposed an epidemic approach to routing in DTNs, in which routing is performed based on absolutely no prior knowledge about the network, and is based only on flooding messages to any encountered nodes. While achieving high delivery, such an aggressive routing technique exhausts the network in terms of bandwidth, buffer and energy, as demonstrated by [Tseng et al., 2002]. Several methods were then proposed to control flooding, often based on limiting the number of replicas used in the network [Spyropoulos et al., 2004, Spyropoulos et al., 2005, Ramanathan et al., 2007].

A second generation of DTN protocols, based on probabilistic routing, was then proposed, in which the history of nodes' encounters was exploited in order to make more informed routing decisions [Lindgren et al., 2004, Sandulescu and Nadjm-Tehrani, 2008, Musolesi and Mascolo, 2008, Boldrini et al., 2007]. Most of these approaches relied on the assumption that a node that has encountered a destination many times or recently is likely to encounter the same destination again. However, this assumption does not always hold for human mobile networks, making the proposed probabilistic routing protocols inadequate in human-based settings. As a consequence, the focus of the research community turned to establishing a better knowledge of human mobility patterns. Models such as Random Walk [Camp et al., 2002] and Random Waypoint [Bettstetter et al., 2003], which assumed each node may equally frequently move and encounter other nodes, were identified as not realistic by various studies [Balazinska and Castro, 2003, Henderson et al., 2004, Hsu and Helmy, 2005, Hui et al., 2005, Spyropoulos et al., 2006], as they did not hold in real-life situations. In response, more accurate mobility models that attempt to cater for human movement characteristics, such as the Community-based Mobility Model (CMM) [Musolesi and Mascolo, 2006] and [Bai et al., 2003, Jardosh et al., 2003, Tuduce and Gross, 2005] were proposed, and the research community focused on collecting and modelling mobility traces, representative of real human movements [Hui et al., 2005, Eagle and Pentland, 2006, Scott et al., 2009, Kim et al., 2006, Rhee et al., 2008, Song et al., 2010]. In [Chaintreau et al., 2007], the authors showed that human mobility exhibits a power-law distribution property of the contact and inter-contact times of nodes. Yoneki et al. [Yoneki et al., 2007b] similarly investigated the properties of human mobility and categorised nodes into four categories based on their mobility behaviour: familiar, familiar stranger, stranger and friend.

Thanks to a better understanding of its properties, mobility was no longer seen as a hurdle to routing; rather, it started to be exploited to help bridge discontinuities in the network. A third generation of DTN protocols, such as [Costa et al., 2008, Daly and Haahr, 2007, Pujol et al., 2009], was started and proved to be effective in delivering content to destinations. For example, properties of social networks have been exploited, in order to assist mobility-based routing [Bigwood et al., 2008, Mtibaa et al., 2008]. Bigwood et al. [Bigwood et al., 2008] exploit Self-Reported Social Networks (SRSN) from users, in order to route messages in the network; they reported a comparable delivery ratio to mobility-based approaches which detect social networks through colocation heuristics.

For future reference and comparison benchmark in the following chapters, we describe next in more details state-of-the-art routing protocols of the lastest DTN generation.

Change in Degree of Connectivity [Musolesi and Mascolo, 2008]. Change in Degree of Connectivity is a probabilistic-based routing protocol, in which carriers are selected based on their degree of mobility. A node with a high *change in degree of connectivity (CDC)* is defined as a node that frequently changes its set of neighbours. This is either due to the node being highly mobile, or placed in a popular place, thus often colocated with many neighbours (e.g., a person collecting a survey in the street). The main principle underlying this routing protocol is that the nodes with higher change in degree of connectivity encounter many nodes, thus making them better carriers for forwarding messages. Given a node h, its CDC value at time t is computed as:

$$U_{cdc_h}(t) = \frac{\mid n(t-\tau) \cup n(t) \mid - \mid n(t-\tau) \cap n(t) \mid}{\mid n(t-\tau) \cup n(t) \mid}$$

where $n(t)$ is the set of nodes that h is colocated with at time t, and τ is a constant time. Upon an encounter between nodes A and B, the messages from A's buffer are sent to node B, if B has a higher *change in degree of connectivity*, or in other words, if B has encountered more distinct users in the time period $t - \tau$ to t:

$$U_{cdc_A}(t) < U_{cdc_B}(t).$$

Social Cast [Costa et al., 2008]. Social Cast builds upon the *change in degree of connectivity* property by introducing a socio-aware component into the routing reasoning. This is done by defining the *colocation to subscribers* of the content value for a node h and a message i, as follows:

$$U_{col_{h,i}}(t) = \begin{cases} 1 & \text{if } h \text{ is colocated with a subscriber of message } i \\ 0 & \text{otherwise} \end{cases}$$

This $U_{col_{h,i}}(t)$ represents the likelihood of node h encountering a destination. The calculated values for $U_{col_{h,i}}(t)$ and $U_{cdc_h}(t)$ are then fed into a Kalman Filter [Kalman, 1960] function, to give an estimate of the overall utility of carrier h at time $t + \tau$ in the future, noted as $\hat{U}_{col_{h,i}}(t + \tau)$ and $\hat{U}_{cdc_h}(t + \tau)$ respectively :

$$U_{h,i}(t + \tau) = w_{col}\hat{U}_{col_{h,i}}(t + \tau) + w_{cdc}\hat{U}_{cdc_h}(t + \tau)$$

where w_{col} and w_{cdc} are weights that adjust the importance of each predicted utility value. Upon encounter, the two nodes exchange their estimated utility for each message (i) that is stored in their buffers and the message is then passed to the node with the higher utility $U_{h,i}(t + \tau)$.

SimBet [Daly and Haahr, 2007]. Daly et al. [Daly and Haahr, 2007] use community ties among individual users, discovered based on social network analysis techniques, in order to estimate the relative importance of users in the network. In so doing, bridge nodes are identified based on their centrality characteristics, and later exploited to deliver content to otherwise disconnected portions of the network.

The authors present a comprehensive proof of the existence of the correlation between the centrality metrics, calculated based on global knowledge of the network, and locally estimated metrics which are based on only a partial knowledge of the network. The routing decisions are then made based on exchanging the pre-estimated *"betweenness"* centrality metrics and locally determined social *"similarity"* to the destination node. The similarity and betweenness are calculated as follows:

- *Egocentric Betweenness*: this metric is calculated using an egocentric presentation of the network, formed when nodes come into contact with each other. The idea of this metric is to identify bridges which help reach disconnected areas of the network.
- *Similarity*: for each node in the vicinity, the number of common social neighbours with the destination is counted. The basis of this idea is that nodes with a lower

degree of separation from a given node are good candidates for information dissemination to that node; the degree of separation is measured by the ratio of common neighbours between individuals in social networks.

Their results indicate an improvement in terms of delivery over preceding protocols such as PROPHET [Lindgren et al., 2004], as well as a reduction in overhead, measured based on the number of forwards each node undertakes. Although we do not directly compare the contributions of this thesis to the SimBet protocol, we will refer to this state-of-the-art protocol through other DTN protocols such as FairRoute [Pujol et al., 2009], which were built upon SimBet to reduce network overhead and achieve fairness.

Online (Self-Reported) Social Network Routing [Mtibaa et al., 2008]. Mtibaa et al. propose a routing protocol for delay tolerant networks based on the information from a user's social network (such as their online Facebook[5] social profile). The authors claim that most properties of links and paths correlate between the self-reported social network and the network discovered based on physical encounters. Therefore, a self-reported social network based approach can be used to disseminate content via paths made of nodes that are socially close (such as friends). Various forwarding techniques based on users' relations in their social networks are then proposed, which define the routing decision that should be taken upon an encounter between any arbitrary pair of nodes u and v. We briefly describe them next:

- Neighbour(k): node u forwards messages in its buffer to v, if u and v are within distance k of each other in the social network graph.
- Destination-neighbour(k): node u forwards messages to v, if v is within distance k of the destination of the message.
- Non-decreasing-centrality: node u forwards messages to v, if v has a higher centrality in the social network graph than node u.
- Non-increasing-distance: node u forwards messages to v, if the social distance from v to the destination of the message is no more than u's distance to the destination.

Each of the described forwarding techniques is then evaluated in a mobile setting. The results present a competitive delivery ratio to that reported by protocols based on discovering the social network through direct physical encounters.

[5] www.facebook.com

Common to all the described state-of-the-art routing protocols is the assumption concerning users ascertained participation in the content dissemination network. Although some of these protocols, such as [Daly and Haahr, 2007, Costa et al., 2008, Mtibaa et al., 2008], attempt to reduce the amount of messages each node is asked to forward, by limiting the number of replica messages distributed in the network, they do not take into account users' interest in routing those messages, nor do they consider the effect that such forwards have on devices' battery. As described in Chapter 1, given the long tail distribution of user's interest observed from research on Web 2.0, users often find themselves contributing in forwarding content that they have no interest in. Furthermore, as we will show in this thesis, reducing the overall network overhead does not necessarily imply that the workload is reduced uniformly and fairly across all nodes in the network. Indeed, it is often the case that, in order to reduce the overall network overhead, protocols look into user's mobility patterns to carefully choose the least number of carriers. However, this reasoning often leads to a small subset of nodes to be repeatedly selected as content carriers over and over again, leading to a highly unfair workload distribution. As these nodes will inevitably see their battery drain very quickly, they are more likely to cease participating in the content delivery network, with detrimental effects on the overall delivery. Therefore, a new approach to disseminating content in delay tolerant networks, in which users participation is not taken for granted, is required.

This thesis aims to close a big gap in DTN research, that of delivering content effectively, while looking at users' participation as the primary concern. We define users' participation in the content delivery network in terms of: firstly, user's interest, that is, whether users care about the content they are asked to forward (Chapter 4); secondly, the amount of local resources, that is, the interested users must have enough resources so to be willing to participate (Chapter 5); and finally, satisfaction, as a way of committing users' resources to the delivery of the content that is valued the most, should there not be enough resources to deliver it all (Chapter 6). Before describing how we reason upon these factors to build our content dissemination protocols, we first present, in the next chapter, a unified model that we refer to throughout this thesis in order to formulate our contributions.

Chapter 3

Model Formulation

As discussed in Chapter 1, human DTNs, formed spontaneously on top of colocated users' mobile devices, can be used to disseminate content amongst users in a community. During the past decade, many DTN routing protocols have been proposed to achieve this goal. However, as derived from Chapter 2, current state-of-the-art solutions for content dissemination fail to cater for the *participatory* nature of these networks. This thesis makes contributions in closing this gap in the field of DTNs, by proposing novel protocols which deliver content effectively and efficiently, whilst not taking users' participation for granted. In order to present our contributions in the upcoming chapters, we propose a unified model describing our target scenario (in Section 3.1 and 3.2), as well as stating the assumptions related to it (Section 3.3).

3.1 Modelling Interest

For content delivery in DTNs to happen effectively, it is important for users who are interested in the produced content to be identified. We require users to have profiles describing their interest and preferences. In a distributed setting, where a central authority does not exist, such profiles can be stored locally on the user's device, thus allowing users to easily manage and update them. Experience from Web 2.0 research shows that users describe their interests in two ways: by using tags or ratings describing *what* they are interested in (e.g., tennis, weekend jogging, etc.), or by explicitly stating *whom* they are interested in receiving content from (i.e., à la Twitter). We refer to the former as *information-centric*, and to the latter as *people-centric* profile. Note that a hybrid model can be used too, where users state both what content they are interested in *and* from whom. Let us describe each model individually, by means of examples from a potential scenario.

3.1.1 Information-Centric

In order to understand the interactions occurring in the spontaneously formed delay tolerant network, let us consider, as shown in Figure 3.1, a scenario where users specify *what* they are interested in. In this scenario, Alice (on the left-hand side of the figure) is interested in receiving content about “Science”, “Sci-Fi”, “Web”, etc. In the right-hand side of the figure, Bob is publishing a video clip which he describes as “Science”. Upon publication, the message is routed via the colocated *mobile network* formed on top of users’ devices, so as to be delivered to users who have described their interest using the same keyword (i.e., Science), such as Alice.

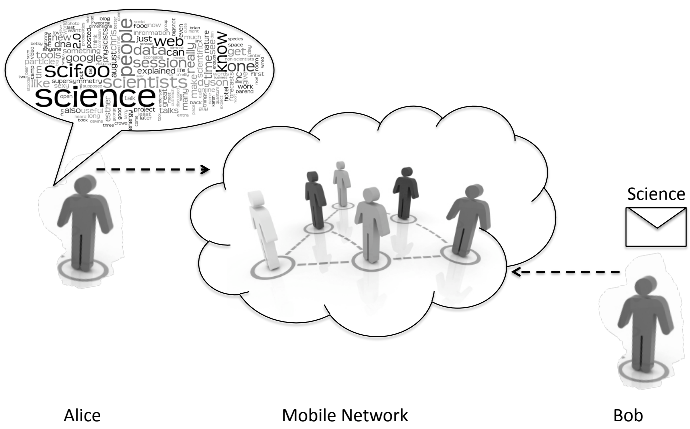


Figure 3.1: Information-Centric Network

Traditionally, to identify interested users in the published content, a user/content matching, based on a universally available taxonomy, is required [Milo et al., 2007]. That is, a universally available taxonomy is assumed to exist, which is used by all users to both describe their profile and categorise their content. In the above example, Alice and Bob require to rely on the same taxonomy, so as to use the same keyword “Science” to describe both their interest profile and the content.

However, experience with Web 2.0 shows that users find taxonomies too rigid and hard to use. Instead, a new trend called *folksonomic* tagging has emerged and is quickly becoming a popular way to describe content and interests within Web 2.0 websites (Flickr[1], MovieLens[2], Delicious[3], etc.). Unlike taxonomies, which over-impose a hierarchical categorisation of content, folksonomies empower users by enabling them to personally and freely create and choose the tags that best describe a piece of information (a picture, a video clip, a URL, etc.). We

[1]www.flickr.com
[2]www.movielens.org
[3]www.delicious.org

expect the same trend to apply to mobile networks, and for users to associate their own personalised tags to the produced content at time of publications. Similarly, the interest profile can be defined either *explicitly* by the user, by entering specific tags they would like to receive content about, or *implicitly* from users' activity, by looking at the tags they used most (i.e., relying on the fact that most people would tag content only if they were interested in it).

To assist us describing an information-centric interest model based on folksonomic tagging, let us refer back to the campus based example we touched upon in Section 1.1, and consider the Film Society at UCL. The Film Society organises weekly movie nights; to advertise such events, a media flyer consisting of a mixture of text and promo video is created by the society managers, tagged with words describing its content, and published to be serendipitously disseminated to interested people around campus. For instance, let us imagine a university student Alice, who is passionate about black and white movies, and has therefore described her interest as a set of tags such as $\{$*cinema, black and white, old*$\}$. Let us assume that this week's Film Society viewing is Hitchcock's "THE MAN WHO KNEW TOO MUCH", and the society manager (Bob) publishes a flyer with tags $\{$*movies, black and white, crime*$\}$.

Based on this scenario, we can formally define a user's interests as follows:

$\forall\, u_x \in N$, where u_x is a user in the network N, $\exists$ a profile P_{u_x} describing user u_x's interest as a set of freely chosen tags, $P_{u_x} = \{t_1, t_2, ..., t_n\}$. Furthermore, $\forall\, t_i \in P_{u_x}$, $\exists$ a weight w_i stating how much user u_x is interested in the tag. This weight could be explicitly defined by the user or implicitly inferred from the usage of tags (i.e., the higher the frequency of the use, the higher the weight).

Similarly, a piece of content c (e.g., a flyer) can be formally modelled as a set of freely chosen tags T_c, describing such content: $T_c = \{t_1, t_2, ..., t_m\}$. Based on these definitions, we can then define user/content matching, which allows us to determine who is interested in what content, as follows: user u_x is defined as a *destination* for content c if $P_{u_x} \cap T_c \neq \emptyset$. In other words, if there is at least a common tag between Alice's interest profile and the advertised flyer (i.e., in this case, the tag *black and white*).

Based on the defined model formulation, for Alice to receive the movie flyer through a mobile content dissemination network, Alice needs to be identified as a recipient of the flyer, through user/content matching reasoning. However, user/content matching is not a trivial task, when tags do not belong to a static, globally defined taxonomy, but are arbitrarily chosen from an infinite personalised vocabulary. For example, in our scenario, Alice may define her interest profile as before $\{$*cinema, black and white, old*$\}$; however, should a flyer be promoted using tags $T_c = \{$*The Big Combo, film noir, awesome*$\}$, a simple user/content matching would fail to identify Alice as a recipient. Therefore, an intelligent user/content matching reasoning system is required, to take into account users' individuality and their diverse vocabulary. We investigate the problem of identifying interested recipients, specifically in the domain of DTNs, in Chapter 4.

3.1.2 People-Centric

In the second scenario, we examine situations where users have prior knowledge about sources of information. Such sources could be people with whom users share an interest, thus they would like to receive content produced by them. For instance, they could represent food or coffee places (e.g., Cafe Nero) advertising their promotions and latest offers for the local community to take advantage of; these promotions can contain pictures of discounted products, along with a barcode to be redeemed. The sources could also be individuals; for instance, they could be college students taking voice recording of the lectures, and sharing them with their fellow classmates.

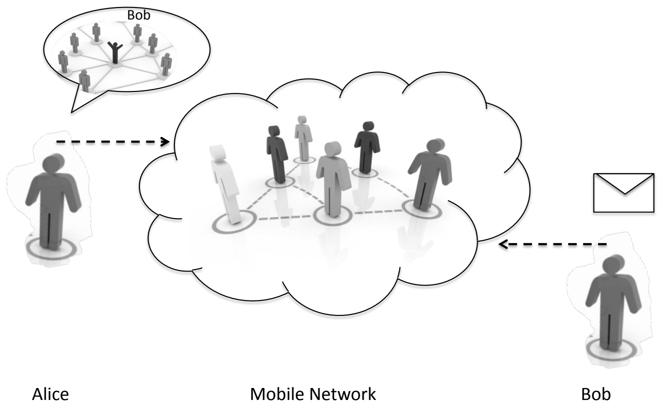


Figure 3.2: People-Centric Network

In such a scenario, users subscribe to specific sources, and in doing so, they opt to receive all the publications from them. The user's interest profile states *whom* they are interested in receiving content from, rather than about what. Figure 3.2 illustrates this scenario, depicting a network in which user Alice has described in her profile, her interest in receiving content from user Bob. In such a scenario, the network first needs to identify Alice as recipient of any content produced by Bob; a message is then routed from Bob, via a network of colocated mobile devices, to Alice. For this to happen, Alice needs to identify her sources of interest. We can model a user's interest in this scenario as a directed graph, where an edge from user u_a (Alice) to user u_b (Bob) means that u_a is interested in receiving content from u_b. The edges can also be labelled with corresponding weights, indicating the strength of the social ties. Such weights can be *explicitly* defined by the user (like in Rummble[4]), or *implicitly* by looking, for example, at the frequency of interaction between users (e.g., in Twitter, the frequency of $@username$ directed messages). Similar to the first scenario, we formally define a user's interest profile as:

$\forall u \in N$, where u_x is a user in the network N, $\exists$ a profile P_{u_x} describing the set of sources whose publications user u_x follows: $P_{u_x} = \{u_a, u_b, ..., u_z\}$. Let c be a piece of content published by user u_s; we can define user u_x as a *destination* for the content c if $u_s \in P_{u_x}$; in other words, if user u_x has explicitly identified their interest in the source of the content.

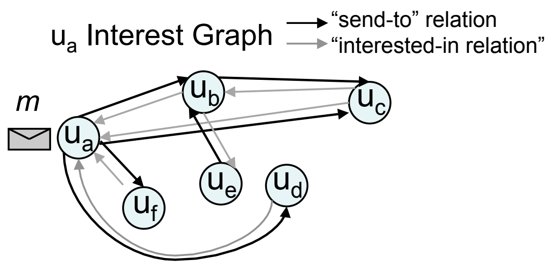
3.1.3 Generalised Interest Model

Although the two approaches of modelling interest (i.e., people-centric and information-centric) differ in various aspects, they can be abstracted into a general model to act as a basis for modelling interest. We define this abstract model as follows: let N present a mobile network of users u_x, each uniquely identifiable. For every node u_x, we define an interest profile P_{u_x} as a set $P_{u_x} = \{t_1, ..., t_n\}$, such that $t_j \in I$ or $t_j \in W$ $\forall j$; where W corresponds to an infinite set (vocabulary) containing all possible words, while I represents all registered IDs in the network. In other words, a node's interest profile is defined either by a set of freely chosen tags, or by the sources that the node is interested in following. As before, users can explicitly or implicitly assign a weight w to each of the profile elements, t_j.

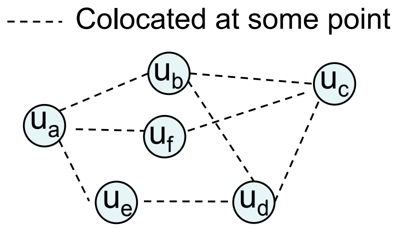
Similarly, for any piece of content c, produced by a user u_s, we assume there exists an associated tagset $T_c = \{t_1, ..., t_m\}$, whereby $t_i \in W$ or $t_i = s$. In other words, T_c is either a set of tags chosen by the source u_s to describe what the content c is about, or it simply is the source's ID. Based on the above general model, user/content matching (i.e., identifying destinations for each piece of content) can be done as follows: u_x is a destination for the content c, if $P_{u_x} \cap T_c \neq \emptyset$.

We can illustrate this general model by means of a directed graph, in which nodes represent users in the network, as illustrated in Figure 3.3a. Based on the abstract model, a directed edge from user u_b to u_a represents the fact that user u_b is interested in the message m comprising of content c (and header h) and produced by user u_a, whether he is interested in all the messages published by the source u_a (i.e., the people-centric case) or specifically interested in this message, because of its descriptive tags (i.e., the information-centric case). When node u_a

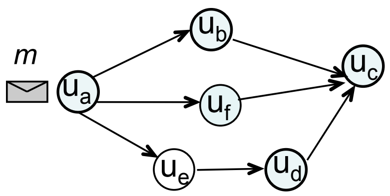
[4] www.rummble.com



(a) Interest Network



(b) Physical Network



(c) Delivery Network

Figure 3.3: Content Dissemination Network

produces some content, the following actions happen: first, the directed edges related to node's interest need to be reversed, allowing u_a to identify destinations, by changing the incoming *"interested in"* arrows, into *"send to"* relations. This is not a trivial task in a distributed setting such as mobile networks, in absence of any global knowledge, and where there is no notion of a central repository to store all profiles. Second, a physical route from u_a to u_b, by the means of multi-hop intermediaries needs to be discovered. This is also not a trivial task, due to the mobility of users and instability of connections. We discuss the challenges faced in discovering delivery routes in the following section, where we model the colocated network of mobile devices. For the purpose of future references, Table 3.1 summarises the unified model presented in this section.

3.2 Modelling Routes

Once destinations are identified in a network, the second challenge is discovering a physical route to reach them. However, as such an end-to-end route does not exist for any given point in

Notation	Entity
u_x	Generic User
P_{u_x}	User's profile
m	Message, $h : c$
h	Header (metadata)
c	Content
T_c	Content description
t_i	Tag

Table 3.1: The Unified Model Notation

time, nodes' future colocations must be predicted to build one. To help us describe this task, let us use the example depicted in Figure 3.3, where user u_a is publishing a message m, comprising a piece of content and any metadata associated to the message (e.g., list of recipients). Figure 3.3b presents the physical colocation amongst nodes; more precisely the dashed lines represent that there was a colocation at some time in the past between a pair node, thus presenting the probability of a future encounter.

Such colocations are not maintained permanently, rather they are driven by human mobility, making the path discovery a very challenging task. For example, let us consider node u_c, which is interested in receiving content from u_a. For this to happen, u_a needs to be aware of future colocations between u_b and u_c, and its own future colocation with u_b, so to build a multi-hop paths lasting during the periods of colocation. This colocation information can be estimated based on predictability and regularity of human movement [Song et al., 2010], as it has been studied and exploited by state-of-the-art DTN protocols [Costa et al., 2008, Daly and Haahr, 2007]. Based on the above example, delivery routes through intermediaries u_b, u_e, or u_f can be discovered to deliver message m from u_a to u_c, as illustrated in Figure 3.3c.

The challenge that arises here is regarding the route that should be selected to deliver message m to user u_c. For instance, in the illustrated example, intermediary u_e may selfishly refuse to participate in delivering message m as he had not requested the content (nor is interested in it). On the other hand, while selecting the path through intermediary u_f guarantees its participation in the sense that it is interested in the message, it does not guarantee the delivery as u_f may not have enough resources to route the message. Thus, a reasoning which takes into account intermediaries' participation, is needed.

3.3 Assumptions

For completeness, we report our assumptions concerning the modelled scenario, users' behaviour and the environment. We assume the modelled scenario takes place in an urban, metropolitan city, such as London, where time, space and interest for casual content production/consumption exist (i.e., as opposed to developing countries). Users maintain an up-to-date interest profile, describing their interest in order to receive produced content in the network. The produced/shared content is delay tolerant in nature and classified as leisure; it is hence not time-critical. Furthermore, the disseminated content covers rich media files, comprising video, picture, and/or music, and is thus bulky in size. Finally, we assume users are not malicious, and do not interfere with the routing process.

In the next chapter, we tackle the first two of our challenges, that is, identifying interested users in an absence of central knowledge (Ch 1), and using content delivery routes that rely on the minimum number of uninterested intermediaries in the network (Ch 2).

Chapter 4

Interest-Awareness

In this chapter, we address the problem of users' participation from the point of view of their *interest* in the content to be relayed. First, we briefly review the state-of-the-art solutions (Section 4.1). We then discuss how user's interest can be determined in a mobile distributed setting, and be exploited by an interest-aware protocol (Section 4.2), which we call *Habit* (Section 4.3). We evaluate Habit by means of simulation. We first describe our simulation settings (Section 4.4), and then report on the results obtained in terms of *precision* (i.e., nodes receive only content they are interested in) and *recall* (i.e., all relevant content is received by interested nodes) in Section 4.5. Finally, we summarise our findings and draw a conclusion regarding this contribution (Section 4.6).

4.1 Background

In human DTNs, disseminating content often involves multiple nodes to act as intermediaries to forward the content from source to destination, catering for the lack of direct contact between the two. In order to select these paths and intermediaries, many DTN protocols have been proposed that aim to maximise message delivery while reducing delay and overhead. To achieve this goal, state-of-the-art protocols have mostly relied on exploiting knowledge from the physical layer, such as estimating probability of encountering nodes, and thus selecting routes with high delivery probability, as we reviewed in Section 2.2.3. What these protocols fail to take into account is information from the application layer, stating user's interests. We claim that this information is particularly important in the domain of participatory networks, where scarce resources on handheld devices impose a restriction on the amount of content nodes can forward before their battery is depleted. As such, it is rational to assume that nodes would be willing to relay content they are interested in receiving in the first place, rather than content they do not care about.

Indeed, as user's interest follows a long tail distribution, it would often be the case for content to pass through many non-interested relayers. State-of-the-art DTN protocols [Costa et al., 2008, Daly and Haahr, 2007, Lindgren et al., 2004, Musolesi and Mascolo, 2008] which reason only on the information from the physical layer would all suffer from overwhelming users, thus threatening their participation. Figure 4.1 illustrates classification of the state-of-the-art protocols in terms of recall (i.e., delivering relevant content) and precision (i.e., avoiding non-interested relayers). We expect the flooding based protocols such as Epidemic [Vahdat and Becker, 2000] to cover the region with high recall (effective delivery) but low precision due to their aggressive forwarding strategy. At the other side of the spectrum, we have social based protocols such as Wait-for-Destination and [Mtibaa et al., 2008], which reason on the interest of those nodes encountered directly. Although such protocols can offer high precision, they compromise on delivery, thus resulting in low recall. Mobility based protocols [Costa et al., 2008, Daly and Haahr, 2007, Lindgren et al., 2004, Musolesi and Mascolo, 2008] fall in the region with average/low precision and average/high recall. This thesis closes the gap in the literature by covering the region with high precision and high recall.

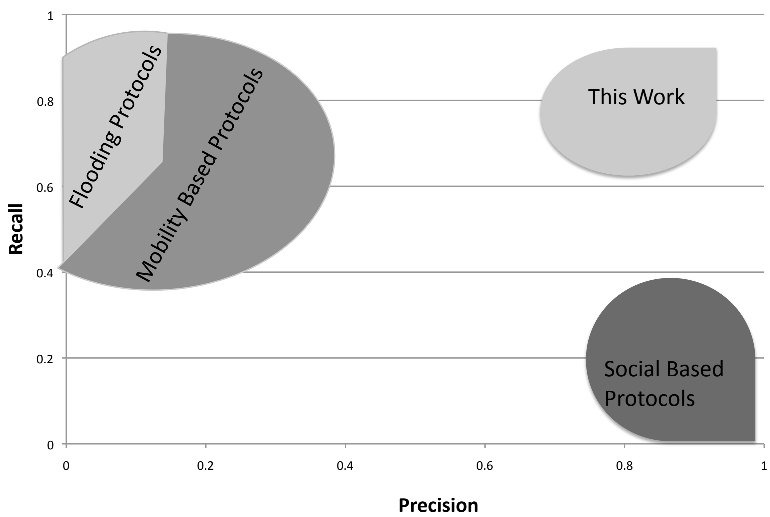


Figure 4.1: State-of-the-art Spectrum

To do so, we propose a content dissemination protocol that exploits information from both the application layer, concerning user's interest, and physical layer, concerning user's connectivity. Using these two distinct sets of information, an approach which builds paths in a source-based manner can be introduced, with publishers identifying the interested recipients at publication time, and selecting paths that reduce the number of non-interested carriers required to relay messages to destinations. To enable a source-based decision making, the interest profiles of users must be dynamically propagated, enabling sources to identify destinations (i.e., reasoning on application layer). Alongside this, the information about users colocation must be logged and processed, in order to learn regularity in human mobility (i.e., reasoning on physical layer). These two layers are hence combined to offer a local view of the *content dissemination network*, enabling sources to build paths that have high delivery probability (based on physical information) while also using the minimum number of non-interested carriers (application information). We describe next a conceptual model offering a solution to the above problem, before presenting a specific implementation.

4.2 Conceptual Model

Our approach to content dissemination is based on the need for a multi-layer model which combines information from the application layer and from the physical layer, as illustrated in Figure 4.2. At the *application layer*, each node specifies the content of his interest as previously modelled in Chapter 3; this can be described either as the sources from whom the user is willing to receive from (i.e., as presented in the application layer of Figure 4.2), or alternatively the topics (e.g., tags) in which he is interested in. At the physical layer, information about nodes' colocation is maintained and updated. These two layers can be leveraged to build a *content dissemination network* (i.e., the middle layer in Figure 4.2), which facilitates a source-based decision making approach to compute paths which cross the minimum number of non-interested nodes. We next detail the information provided by each layer, the combining process of the two layers, and the path selection reasoning.

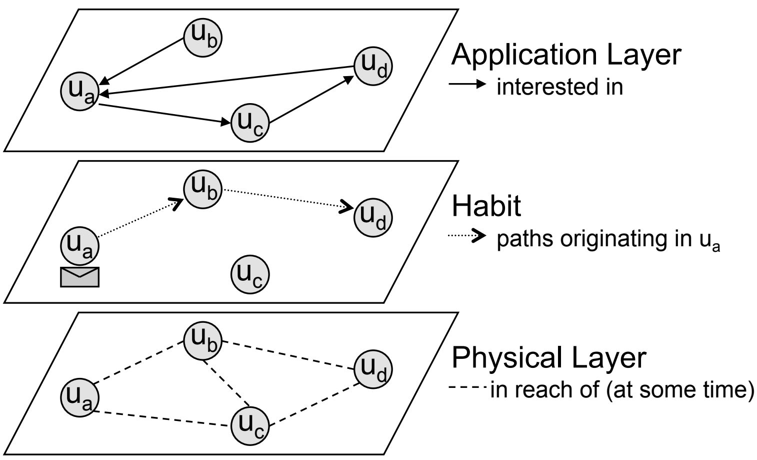


Figure 4.2: A Multi-layer View of the Conceptual Model

4.2.1 A Dual Approach Mode for Modelling Users Interests

One of the main components of any content dissemination network is identifying who the recipients of messages should be. As we discussed earlier in Chapter 3, in practice, user's interest can be modelled in two different ways: either by describing a social network of who is interested in whom (i.e., *people-centric*), or by associating a set of keywords describing who is interested in what (i.e., *information-centric*). Consequently, matching published content to users happens either via social network reasoning or via folksonomic reasoning.

People-Centric

In this model, at the application layer, each user specifies the sources from whom they are willing to receive content. As seen in typical Web 2.0 applications, such sources could be people with whom I share music tastes (e.g., Last.fm[1]), and thus I like to receive a music clip which they regard as interesting. They could be people with whom I share food tastes (e.g., Rummble[2]) and whose publications about newly opened restaurants and/or ongoing offers I would like to receive, and so on.

In this scenario, we assume each user has an up-to-date profile describing his interests in terms of his social network. In essence, such social network can be thought of as a directed graph where nodes are users and edges reflect the interest relation (e.g., an edge from u_b to u_a means that u_b is interested in receiving content from u_a, as we previously modelled in Section 3.1). The matching between published content and user is then simply done based on matching the ID of a message publisher with a user's interest profile.

Information-Centric

For scenarios where producers and consumers do not know each other, an information-centric interest model can be used, in order to let users describe what they are interested in, rather than in whom. Content is thus described by a set tags T_c, which are then used in user/content matching as modelled in Chapter 3.

While being extremely easy to use, folksonomic tagging has often been criticised for lowering the efficacy of matching [Scott and Bernardo, 2006], resulting in more *undiscovered* destinations (i.e., interested users are not identified due to different choices of tags for describing their profile). This is due to the number of synonyms, homonyms, polysemy, as well as the unavoidable heterogeneity and individuality of users. Therefore, while user/content matching in the people-centric model can be simply accomplished as string comparisons over publisher's ID, the same cannot be said for a folksonomy-based information-centric model, where it is

[1]www.last.fm

[2]www.rummble.com

crucial to exploit further matching as tags in users' profiles are not simply pre-defined content categories, but freely chosen words.

Indeed, tag matching in folksonomy is a well-known problem in Web 2.0 websites, and various algorithms have started to be put forward to tackle it via a variety of tag expansion mechanisms [Zanardi and Capra, 2008, Bertier et al., 2009]. Put it simply, the common underpinning idea is to maintain a global tag correlation matrix M, which keeps track of what tags are being used in conjunction with what other tags (e.g., to describe pictures, videos, blog posts), and how often. For example:

	t_1	t_2	t_3	t_4	t_5	$\ldots$
t_1	-	3	12	4	1	$\ldots$
t_2	3	-	2	8	0	$\ldots$
t_3	12	2	-	0	0	$\ldots$
t_4	4	8	0	-	0	$\ldots$
t_5	1	0	0	0	-	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	-

When a user is searching the Web 2.0 website with query tags say t_1 and t_2, the tag correlation matrix is consulted first, to identify the tags that are most related to the query tags (in the example, t_3 and t_4). Content that has been tagged with any of the expanded set of tags t_1, t_2, t_3, t_4 is deemed relevant, and thus retrieved and subsequently ranked. The rationale is that if t_1 and t_2 have been very frequently used together with t_3 and t_4 on some pieces of content, then there is a high probability that t_3 and t_4 are alternative words to t_1 and t_2, and people may use them instead (e.g., car racing and Formula 1). To reduce the risk of undiscovered destinations, tags are thus expanded prior to routing. We argue that a similar approach could be applied in our scenario too. However, we cannot rely on a centrally available tag correlation matrix, dynamically built from the tag usage made by all users in the system, and ready to be queried by anyone, anywhere and anytime; rather, a fully distributed approach is required, whereby each device relies on locally available knowledge to perform tag expansion. In this regard, each node dynamically maintains a local tag correlation matrix M which counts how many times each pair of known tags was used together to describe a piece of content. The matrix is updated in two occasions: upon message creation (thus by the source), and message reception (thus by the carriers and destinations). Upon message creation, the source node updates its local M by increasing the entry $M[i, j]$ for each pair of tags t_i, t_j the user associated to the message; symmetrically, whenever a node receives a message, either to forward or because it is the final recipient, the header of the message is consulted so that the local matrix M is updated in the same way, based on all tags that are attached to the message describing its content. Note that M is a triangular matrix, and also very sparse, so its storage overhead is low; should M grow excessively, entries with very low counters could simply be removed.

The described tag correlation matrix is then consulted, so that the k tags most related to those in T_c are selected and added to the *expansion set* T'_c. Note that other strategies are possible when computing the expansion set; however, in this thesis we focus on the k-Nearest Neighbour [Cover and Hart, 1967] strategy applied on a per-tagset basis. Possible alternatives are kNN applied to each individual $t_i \in T_c$, rather than to the whole tagset at once; also, rather than expanding using the top k most related tags, thresholding could be added, so that only the top k related tags, whose relation to the original tag is above a given threshold, are considered. We leave these possible alternatives as part of our future work, which we will describe in Chapter 7. Note also that, once tags in T_c have been expanded, leading to the construction of T'_c, the process could be iteratively repeated for all tags in T'_c, in an attempt to uncover an even bigger set of tags related to T_c (i.e., those originally used by the creator of the message). The expanded tagset T'_c is then used to identify user u_x as a recipient of the message if $T'_c \cap P_{u_x} \neq \emptyset$; in other words if the user has at least one tag in common between his interest profile and the new expanded message tagset.

For both information-centric and people-centric models, we assume availability and accuracy of user's profile on his own device. We then require a distributed approach to handle profile propagation, so to enable the source of content to identify the interested nodes, as we will discuss in Section 4.2.3.

4.2.2 A Prediction-Based Model for Users' Mobility

Apart from determining recipients, nodes must have knowledge about future encounters in the network, so to build routes to the recipients in a multi-hop fashion. Although 100% accurate information cannot be provided, good estimations can be computed based on heuristics. Indeed, mobility prediction has been extensively used by DTN protocols, exploiting the fact that human mobility is not random, but is driven by routines that are predictable [Song et al., 2010, Rhee et al., 2008].

Likewise, we rely on a prediction-based technique to assist nodes to gain a local view of the colocations in the network. Let us describe the construction of this layer from a single node's perspective. Information on direct encounters are monitored and the time for each encounter (the hour of the day and the day of the week) is recorded. A week period is divided in 7 days, and each day in 24 hour slots; whenever an encounter occurs, the corresponding entry is updated, and this logging continues *week after week*.

Based on these collected logs, nodes can now compute what we call a *regularity weight*, that is, the number of times a node has met another node in a given slot of the week, over a certain observation period (e.g., last five weeks). For instance, if node u_a has met node u_b four times in the last five weeks (observation period) on Mondays in the hourly slot 10AM-11AM, then the regularity weight between u_a and u_b for such day/time slot is set to 0.8. As we will describe in Section 4.2.3, the regularity weight can then be used in path discovery, enabling the source node to find highly probable routes to destinations within the time-to-live (TTL) of the published content.

Considering the sparsity and diversity of encounters in an urban scenario, it is important to rely on the most repeatedly-seen nodes, discarding any further reasoning on nodes who have only been met once or so (i.e., we refer to these nodes as "strangers" [Yoneki et al., 2007b]). In so doing, we define *familiar strangers* as those nodes that node u_a regularly meets (e.g., travelling to/from work, while at work, living in the same neighbourhood). Node u_a and u_b are familiar strangers if they were colocated frequently enough according to some parameter. We then define u_a's *regularity table* as the set of all regularity weights computed for all his familiar strangers.

To limit overhead, each node maintains regularity information about a maximum number of familiar strangers only (we refer to this parameter as *maxFS*), giving priority to nodes that are most regular and with whom we share more interests. To reduce overhead further, the regularity interval can be chosen to be more coarse-grained, thus covering human-meaningful time slots rather than fine-grained hourly slots (e.g., commute time slot, working hours slot, etc.).

4.2.3 Building the Content Dissemination Network

In order to build the content dissemination network, two sets of information are required: information from the application layer, specifying who are the recipients of the produced message (Section 4.2.1), and information from the physical layer, specifying how to reach the identified recipients (Section 4.2.2). This information can then be combined and used for the purpose of path selection. Each node exchanges its regularity table with his familiar strangers upon encountering them, and this is repeated up to a certain number of hops in the network (*maxHops*). Therefore, over time, each node possesses a partial view of what we refer to as *regularity graph*, that is, a directed graph where an edge from u_a to u_b means that u_b is a familiar stranger to u_a; the edge is then labelled with the corresponding row in the regularity table, detailing the probability of u_a meeting u_b in any given time slot (i.e., the estimated regularity weight for any given time slot in the week). To avoid unnecessary exchanges and processing of metadata, nodes time-stamp the last update made to their regularity graph so that, upon meeting a familiar stranger, regularity tables are exchanged only if a certain amount of time has elapsed since the last synchronisation.

Besides propagating regularity tables, familiar strangers also propagate their interest networks. This is done by nodes exchanging their interest profiles up to a certain number of hops (*maxHops*), upon encountering familiar strangers. Based on the gathered interest profiles, a partial view of what we call an *interest graph* is formed, where by interest we mean either interested in the tags of the published content (information-centric) or the publisher himself (people-centric).

4.2.4 Reasoning on the Content Dissemination Network

When node u_a publishes a message m, the following three steps are performed: first, recipients are determined; second, candidate paths, that would enable a message m to reach such destinations, while also crossing the minimum number of non-interested intermediaries before m expires, are calculated. Finally, the path with the highest probability of delivery is selected. We describe these stages by means of an example next.

Determine Recipients

At the time of publication, the source associates a set of tags, $T_c = \{t_1, \ldots, t_m\}$, to the message describing the produced content, or the source's ID (for information-centric and people-centric respectively). The collected user profiles are then consulted, in order to allow the source to determine the *known* destinations. Note that this list may not be complete, as there may exist interested nodes that the publisher is unaware of, either due to the publisher's partial view of the interest network, or because of mismatches among tags. Therefore, it is important to design the protocol such that nodes can opportunistically forward the message, should they encounter any originally *undiscovered* destinations.

Find Cheapest Paths

For each known recipient, the locally maintained regularity graph is consulted, to find all those paths that would enable message m to reach such destination before the message itself expires, while following the *cheapest* routes. By cheapest, we refer to those routes that minimise the number of uninterested intermediaries that have to be relied upon. The cost associated to a path is computed simply as the number of intermediary nodes that would have to carry a message they are not interested in.

With reference to the example in Figure 4.3, let us assume that u_a publishes m on Monday 10AM and that m must be delivered within the next 48 hours. Let us also assume that the

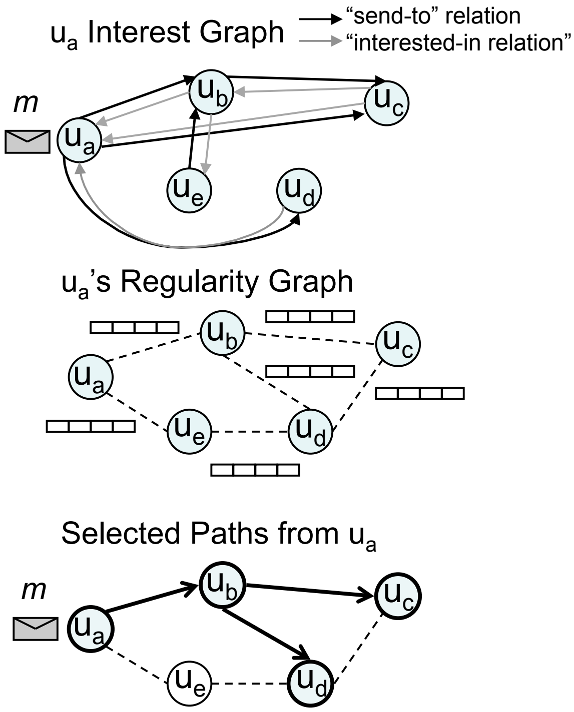


Figure 4.3: Content Dissemination Network from u_a's View Point

regularity graph tells u_a that u_d can be reached before the message expires along any of these routes: $u_a \rightarrow u_b \rightarrow u_d$, $u_a \rightarrow u_b \rightarrow u_c \rightarrow u_d$, and $u_a \rightarrow u_e \rightarrow u_d$. While the first and second routes have a cost of 0 (i.e., no uninterested carriers have to be relied upon), the third has a cost of 1, and it is thus discarded at this stage. While exploring the regularity graph to find routes, a delivery probability is also computed for each path, as the minimum regularity weight associated to any of the crossed edges. To reduce the computational complexity of this stage, we have adopted a simple heuristic: given a path $X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_n$, only the *first* non-zero regularity weight between a pair of nodes X_i, X_{i+1} is considered (provided that the associated time slot is between the current time slot and the message expiry slot), before moving on to the next edge X_{i+1}, X_{i+2}. The delivery probability associated to a path is thus an underestimate of the actual one. With reference to Figure 4.3, the path $u_a \rightarrow u_b \rightarrow u_d$ could exhibit (for instance) regularity weight between u_a and u_b of 0.7 on Mondays 2-3PM, and regularity weight between u_b and u_d of 0.3 on Tuesdays 9-10AM; the overall delivery probability following this route would thus be $\min\{0.7, 0.3\} = 0.3$.

Select Paths

For each destination, if more than one path had been computed at the previous stage, the one with the highest delivery probability is now selected. The whole set of source-computed paths is then associated to m, and routing starts once the next hop in any such paths is encountered. Upon such encounter, a copy of m is passed on, together with the source-computed paths in full. A message will remain on a node (be that the source or an intermediary) until either all its next hops in the computed paths have been encountered (and they have thus received a copy of m), or the message expires. In so doing, we enable opportunistic delivery, as we discuss later.

Once computed, source-selected paths are not changed so that relayers do not have to re-run the above processing but simply follow what is stated in the message header. However, as we discussed earlier, the source may not always be aware of all nodes interested in receiving its messages, as the interest network is propagated up to certain number of hops only, and due to the sparsity of folkosonomic tagging. Intermediaries may thus check if there exists any node u_x in their view of the interest graph that is willing to receive messages from u_a but that has not been included in any source path yet (information that is carried along with m). If that is the case, the intermediary may then follow the same steps that u_a did, and enrich m's header with new paths to deliver m to the newly discovered destinations too.

Furthermore, intermediary nodes act opportunistically and deliver messages to interested nodes, should they encounter them directly (i.e., accounting for non-certainty of the prediction technique). This is done by intermediaries performing a user/content matching between the profile of the encountered user and the content they are currently carrying.

4.3 Realisation

We now summarise the realisation of the proposed content dissemination approach. We refer to this realisation as *Habit*, and present its pseudo code next.

u_a: source node, R: set of recipients

$Paths$: The set of the cheapset paths from u_a to all $u_d \in R$

$Neigh[u]$: The direct neighbours of node u in the regularity graph.

$Reg(u, v, t_{now}, t_{exp})$: first non zero regularity weight (w_{reg}) between nodes u and v occurring at time t, $t_{now} \leq t \leq t_{exp}$. Returns -1 if no non-zero w_{reg} exists before t_{exp}.

FindCheapestPaths (u_a, *R*, t_{now},t_{exp})

1: $Paths = \emptyset$;

2: for all $u_d \in R$ {

3: $current.path = \emptyset, current.cost = -1,$

4: $current.minReg = \infty, destMinCost = \infty$

5: $RecursePaths(u_a, u_d, current, destMinCost)$

}

$RecursePaths(u, d, current, destMinCost)$

if $((u \notin current.path) \ \wedge \ (destMinCost \geq current.cost))$ {

  $current.path = current.path \cup \{u\}$

  if (u==d) {

    $destMinCost = current.cost$

    $Paths = Paths \cup \{current\}$

    return

  }

  if $(u \notin R)$ $current.cost++$

  for all $v \in Neigh[u]$ {

   w_{reg}=$Reg(u, v, t_{now}, t_{exp})$

   if $(w_{reg} > 0)$ {

     if $(w_{reg} < current.minReg)$

      $current.minReg = w_{reg}$

     $RecursePaths(v, d, current, destMinCost)$

   }

  }

}

SelectPaths (R)

for all $d_i \in R$

  $max\_reg = 0$, chosen_path

  for all $p_i \in$ Paths s.t. $p_i.destination = d_i$

     if $(max\_reg < p_i.min\_reg)$ then

     $p_i.min\_reg = max\_reg$ and

     $chosen\_path = p_i$

  $\{ChosenPaths\} \leftarrow chosen\_path$

We now proceed with Habit's evaluation. We first describe our simulation settings in Section 4.4, before presenting the results of Habit's performance in Section 4.5.

4.4 Simulation Settings

In order to evaluate this contribution, we thoroughly assessed Habit by means of simulation. We used the ONE simulator [ONE, 2010], simulating a realistic network environment by using real traces which we describe in Section 4.4.1. In Section 4.4.2, we define our metrics, before introducing our benchmark protocols which act as upper-bounds for the defined metrics, in Section 4.4.3. We then list our parameter settings in Section 4.4.4.

4.4.1 Datasets

To perform a realistic evaluation of our proposed content dissemination protocol, we needed a dataset that combined information about people's movement in an urban setting, together with details of their interest described as either social network (i.e., for people-centric), or as database of tags (i.e., information-centric).

Mobility Traces: Reality Mining Dataset [Eagle and Pentland, 2006] is a well-known set of mobility traces, which has been used extensively within the DTN community. In terms of mobility, Reality Mining traces contain colocation information from 96 subjects at the MIT campus over the course of the 2004-2005 academic year, to whom Bluetooth-enabled Nokia 6600 phones were given; colocation information was collected via frequent (5 minute) Bluetooth device discoveries.

To work with a more manageable dataset, we performed experiments using various 3-month portions of the whole dataset. The results reported in the next section refer to the period September-December, when students are around the campus, while still including the occasional holidays. Of these 90 days, the first couple of weeks are used as training (as we will state later when discussing parameter setting, Section 4.4.4), to learn the familiar strangers and the interest network. After this period, nodes start publishing content as we will describe next.

Interest Network: In terms of interest network, the Reality Mining traces include information about phone activities of the participants. Hence, a social network can be implicitly extracted by looking at the exchanged voice and text messages, creating an edge in the social network between users who made calls to each other and/or exchanged text messages [Lindgren et al., 2006]. For the purpose of our evaluation, we assumed that the friendship relation subsumes the interest relation, that is, if users u_a and u_b are linked in this inferred social network, then they are also interested in receiving content they produce; similarly to Web 2.0 scenarios, a pre-analysis of this social network reveals a power-law degree distribution. Finally, we have mapped this social network to the mobility traces, preserving users' identity. To model content publication, a user is then picked at random every hour and a message is injected to the network. Note that one publication per hour is very light and perhaps not a realistic case as it does not represent the distribution of users' activity over the course of a day (i.e., day vs. night). However, our focus here is on delivering produced messages within their time-to-live instead of distributing the produced load. Moreover, as we describe later, we extend our evaluations so to include a more realistic model for publication rate based on a real dataset.

The inferred social network is only suitable for evaluation of Habit with a people-centric interest model, whereas for an information-centric model a dataset which describes the individual interests of users in terms of tags is required. To cater for these scenarios, we have used MovieLens [MovieLens, 2010], which contains information gathered from the homonym movie recommendation website. The dataset consists of tuples (i.e., records), containing: movie identifier $movieId$, user identifier $userId$, a set of $tags$ (i.e., words or short sentences) that the user associated to that movie to describe it, and a $timestamp$ of when the user stored such record in the website.

We used MovieLens data to simulate both users' profiles and message publication. More precisely, each user's profile is built as the set of tags the user has *ever* used to tag movies in the dataset; we worked here on the assumption that users would tag movies for which they have an interest. Furthermore, each record $[movieId, userId, tags, timestamp]$ in MovieLens has been converted to a message publication event in our simulation, where the published message itself consists of $[userId, tags, timestamp]$. These $tags$ (i.e., T_c) are then expanded and used to compute message paths, as previously detailed in Section 4.2.1. Note that field $movieId$ is irrelevant (in practice, it would be the actual media content being shared). The last field, $timestamp$ (ts), could be used to simulate the rate of publication; we postpone the description on how we do this task to later, where we describe the publication rate in details.

The MovieLens dataset contains 15,240 distinct tags, used a total of 95,580 times, and applied to 7,601 distinct movies by 4,009 different users. These users produce 55,484 publications in around 90 days time (which perfectly matches the time window considered for the colocation traces). In order to sample 96 users from the 4,009 present in the dataset, we restricted our attention to those users who have tagged at least 20 movies; moreover, we filter out those tags that have been used less than 5 times overall. The resulting dataset still shows the common behaviour seen in Web 2.0 applications: that is, 20% of the users publishes 80% of the messages [Cha et al., 2007]. From here, we finally chose the 96 MovieLens users to be paired to the 96 Reality Mining ones so that such distribution was maintained. We then overlaid the extracted MovieLens dataset to Reality Mining users at random. The results shown in the next section illustrate averages of 20 runs of random overlaying.

Let us now go back to the postponed issue of publication rate. Our pre-analysis of the MovieLens dataset illustrates that time stamps are highly clustered in very short periods of time, as if users rated a fairly large set of movies all at once. This behaviour is peculiar to the dataset at hand, and not really representative of content production rate. In order to mimic a realistic content publication rate, we discarded the original MovieLens $timestamps$ and replaced them with timestamps taken from Digg [Digg, 2010], the content bookmarking website. We chose the Digg dataset as it is an example of how users consume digital content, and allows us to realistically model users activity, in terms of both the frequency of consuming content (i.e., publication rate) and the time of doing so (i.e., publication timestamps). We sampled 96 Digg users in the same distribution-preserving manner adopted to extract the 96 MovieLens users. Note that the pairing of Digg users and MovieLens users was not done at random; rather, we ranked Digg and MovieLens users by number of publications, and paired them by matching their rank. Therefore, publications made by a user in MovieLens had their timestamps replaced by those made by the corresponding Digg user. In this manner, we are maintaining active users across the network, such that the users who publish many messages are given a higher publication rate to match their need.

4.4.2 Metrics

The goal of our content dissemination protocol is to enable *efficient* content dissemination without compromising on *effectiveness*. That is, relevant messages should be received by interested nodes, while minimising reliance on uninterested carriers. In order to quantify the extent to which Habit achieves this goal, we have computed *precision* and *recall*, two widely used measures for evaluating the quality of results in Information Retrieval [Manning et al., 2008].

Precision is defined as the number of relevant documents retrieved by a search, divided by the total number of documents retrieved by that search. In the domain of content dissemination, we use precision as a measure of efficiency, and we compute it as the ratio of relevant messages received by a node, out of all messages it received:

$$Precision = \frac{\{RelevantMsgs\} \cap \{ReceivedMsgs\}}{\{ReceivedMsgs\}}$$

Intuitively, the higher the precision, the higher the efficiency of the protocol, as it minimises the number of irrelevant messages received by a node.

Recall is defined as the number of relevant documents retrieved by a search, divided by the total number of existing relevant documents (which should have been retrieved) instead. In our domain, we use recall as a measure of effectiveness, and we compute it as the ratio of relevant messages received by a node, out of all relevant messages published in the system:

$$Recall = \frac{\{RelevantMsgs\} \cap \{ReceivedMsgs\}}{\{TotalRelevantMsgs\}}$$

Therefore, the higher the recall, the higher the effectiveness of the protocol, as it maximises the number of relevant messages received by a node.

4.4.3 Benchmarks

We have compared the level of precision and recall that Habit achieves, with two benchmarks: *Epidemic* and *Wait-for-Destination* which are the most effective and most efficient DTN protocols respectively.

Epidemic - Upon receiving a message, each node stores it locally; whenever the node comes in proximity of other nodes, a copy of the message is sent to them, regardless of their interest, until the message expires. This algorithm is expected to reach the best performance in terms of recall. However, it will do so at the expense of precision. Due to resource limitations and/or users' uncooperativeness, we do not expect this protocol to be applicable in the scenarios we are focusing on; however, it does provide a benchmark in terms of the best achievable recall.

Wait-for-Destination - At the opposite end of the spectrum, we consider a protocol that does not rely on any carrier. Rather, the publisher of a message holds a copy of it until it expires; whenever it encounters directly the nodes that are interested in its messages, a copy is passed. While obviously representing the worst case scenario in terms of effectiveness, the protocol is also expected to exhibit the highest precision (i.e., 100%), as messages are only passed to interested nodes. This protocol could be used in very uncooperative scenarios, where selfish nodes are not willing to use their resources to route messages for others, and in cases of very frequent encounters, where multi-hop routing is thus less necessary.

Moreover, we compare Habit against the major state-of-the-art content dissemination DTN protocols, which we previously described in detail in Chapter 2. These are *SocialCast* [Costa et al., 2008], *CDC (change in degree of connectivity)* [Musolesi and Mascolo, 2008], and *Neighbour(k)* [Mtibaa et al., 2008]. Given that these protocols, particularly *SocialCast* and *CDC*, were not originally designed for our metrics (in particular, precision), we have redefined some of their behaviours in order to make a fairer comparison with Habit. We next describe these required adjustments, as well as the creation process of the *friendship network* that is required by the Neighbour(k) protocol. Unless stated otherwise, parameters associated to each protocol are as advised in the original papers.

To put a cap on the aggressive nature of CDC and its derivative SocialCast, and increase their chance of competing with Habit in terms of precision, we redefine the algorithm such that local copies of messages are deleted after the messages are forwarded (i.e., the replica variable is set to zero in Social Cast). However, this is done with the exception of the messages at the source: the local copies of those messages remain at the source (until the messages expire), enabling the source to incorporate opportunistic forwarding at all time (i.e., the very same reasoning exists in Habit protocol too).

In order to apply the *Neighbour(k)* protocol to our simulation environment, we required a self-reported social network of users so that routing can happen based on social network reasoning (as previously discussed in Section 2.2.3). More precisely, we need a social network presenting users' *friendship network*, as well as an interest network describing users' interest. To cater for the latter one, we used the inferred Reality Mining social network to model users' interest as previously described in Section 4.4.1. To cater for the former network, given that the Neighbour(k) protocol routes messages by reasoning on the social links, we require the friendship network to have more links than just those of the interest network; this is to avoid nodes routing messages only upon direct encounters to the destination (i.e., Wait-for-Destination protocol). We thus build a friendship network, by assuming that the edges in the interest network are a subset of the edges in the friendship network. In other words, a user can have many friends without particularly sharing interests with all of them. To model such a friendship network, we have thus started from the interest network at hand, and expanded it so to create a more denser network containing more edges amongst users. In so doing, we have relied on the recent research [Liben-Nowell and Kleinberg, 2007, Dell'Amico and Capra, 2008], tackling the problem of missing links in social networks. For a given social network, these approaches discover other potential links amongst users based on properties of the network. In particular, we rely on the approach proposed by [Dell'Amico and Capra, 2008], and feed the

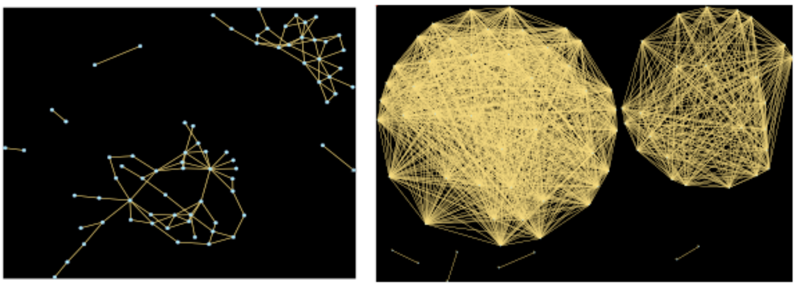


Figure 4.4: The Social (interest) Network and the Inferred Friendship Network

existing (inferred) Reality Mining interest network as the input to the Personalised Page Rank algorithm [Dell'Amico and Capra, 2008], which in return provides us with a denser *friendship* network, by means of link propagation. Figure 4.4 illustrates the structure of the interest network (before propagation) and the inferred friendship network (after propagation); the latter is the one being used for routing in Neighbour(k).

4.4.4 Parameters

Table 4.1 summarises the parameters used to evaluate Habit. We use a training period of 35 days (that is, 5 weeks), during which nodes log their colocation information, and learn more about the network by exchanging their interest profiles and regularity tables. We divide the 1-week logging period into slots of 4 hours each (that is, 42 slots per week). The divided regularity tables contain information for at most 10 familiar strangers (maxFS) and are propagated for up to 4 hops away (maxHops). We next present the results of the sensitivity analysis performed on Habit both for people-centric and information-centric cases, which helps us draw an explanation as to the values assigned to the main parameters.

Simulation Duration	90 days
Training Period	35 days
Regularity Interval	4 hour time slots
maxFS	10
maxHops	4

Table 4.1: Habit Simulation Parameters

4.5 Results

4.5.1 Sensitivity Analysis

In this section, we present the experiments performed in order to help us quantify the effect of various parameters of the Habit protocol, and guide us in tuning them accordingly.

We first tune Habit's fundamental parameters, such as training period and maxHops. We perform this tuning while using a people-centric implementation; the resulting parameter values are equally used in the information-centric approach. We then concentrate on setting information-centric parameters only, as used in expanding tags to discover destinations.

maxHops. In order to analyse the effect of the maxHops parameter of our protocol (i.e., how far the interest profiles and regularity tables should be propagated), we have analysed the relationship between the colocation network and the interest network. We found that the vast majority of users are only a few physical hops away from users to whom they are directly connected in the interest network. Figure 4.5 illustrates the cumulative distribution of the distance, measured as the number of hops in the colocation network between nodes directly connected in the social network. As shown, 94% of the users are within 4 hops away from producers of relevant content; this means that our source-based routing approach is well justied in this scenario, as content producers can easily gain enough knowledge to compute the full route that messages should follow to reach interested destinations without causing too much protocol overhead, by setting the maxHops parameter to 4 (we have done so in all the remaining experiments).

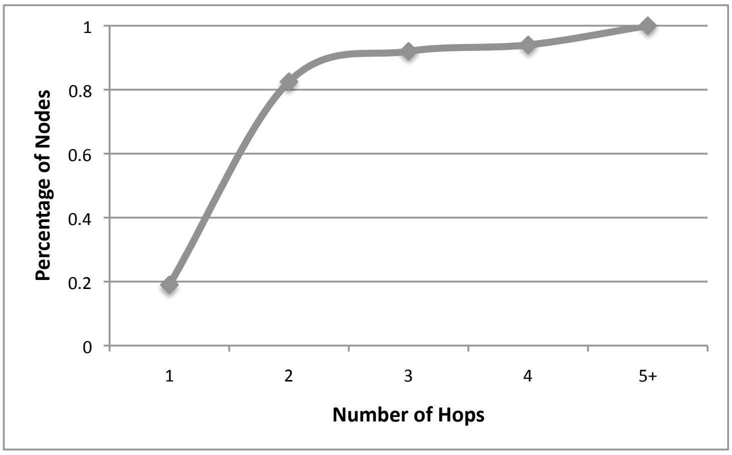


Figure 4.5: Cumulative Distribution of the Distance between Content Producers and Interested Consumers

Training Period. In order to assess the effect of training period on our protocol, we introduced a variant of our protocol to act as upper-bound, which we refer to as Oracle. Oracle differs from Habit regarding the discovery of the interest network. More precisely, in Oracle we assume nodes have complete a priori knowledge of the interest network (i.e., who is interested in receiving content from whom), while they still need to gradually learn their familiar stranger network. In so doing, we can single out the two dimensions of the problem (i.e., learning the regularity graph and learning the interest network), and separately evaluate their impact. It is worth noting that the described Oracle is a manipulated protocol to work as our benchmark comparison and it would not exist in reality due to the distributed setting of mobile networks.

We have performed the following experiment, where we have set the time-to-live of messages to 20 days, and we have then varied the duration of the training period from one week to eight weeks. In so doing, we have studied the impact that learning has on recall (i.e., how long it takes to achieve good delivery ratio); more precisely, with Oracle, we have studied the impact of learning the familiar stranger network (while knowing the full interest network); with Habit, we have studied the impact of learning both the familiar stranger network and the interest network. Figure 4.6 presents these results.

Figure 4.6: Effect of Training Period on Recall

Let us consider Oracle first: even if the social network is fully known, routes connecting content producers to interested recipients cannot be found in the familiar stranger network. It takes about three weeks to learn the familiar stranger network to a good extent, after which high recall can be observed. The same trend can be observed for Habit: in this case, interested destinations have to be learned as well, and after a period of about 5 weeks, the recall of the two approaches converges, thus demonstrating that interests can be propagated via a small number of hops (in the experiment, 4 hops). We have thus set the training period to 35 days (that is, 5 weeks) in all remaining experiments.

Tag Expansion *k*. We now turn our attention to information-centric specific parameters. We evaluate the suitability of the proposed distributed tag expansion approach to retrieve relevant content while varying parameter k, which corresponds to the number of tags being expanded, $k \in \{5, 10, 15\}$. We also test the accuracy of our protocol with and without recursion, that is iteratively repeating the tag expansion process for all already expanded tagset T'_c. The rest of parameters are the same as previous experiment, with the exception of messages time-to-live being set to four days. The outline of this experiment is as follows: we begin by playing the traces for a 35 day training period, during which nodes learn about each other's regularity of movement and interests. After this training period, the following happens: upon message publication, the source node does not include all the tags T_c it would normally associate to the content. Rather, it drops a 50% random subset ($droppedTags$) of the message's tags. After this action, our proposed distributed tag expansion technique expands the remaining tags ($T_c \setminus droppedTags$) in the way that was previously described in Section 4.2.1. The aim of this experiment is to investigate the ability of our technique to effectively recover *tags* and *destinations*. More precisely, *tags' recovery* computes the proportion of the dropped tags that could be recovered by tag expansion:

$$\frac{droppedTags \cap T'_c}{droppedTags} \in [0, 1]$$

Note that, even if some tags are being dropped, the remaining ones may still be sufficient to identify all interested recipients if their profiles P_{u_j} contain at least one tag $t_i \in (T_c \setminus droppedTags)$. To quantify the importance of our approach in recovering destinations (i.e., users) which would have become unreachable otherwise (i.e., $P_{u_j} \cap (T_c \setminus droppedTags) = \emptyset$), we have also measured *destinations' recovery*, that is, how many message's recipients have been recovered purely thanks to the expanded tagset:

$$\frac{(dest(T_c) \setminus (dest(droppedTags)) \cap dest(T'_c)}{(dest(T_c) \setminus (dest(droppedTags))} \in [0, 1]$$

Figure 4.7 illustrates the results of destination and tag recovery while varying the k parameter. The first important observation is that tags' recovery is much lower than destinations' recovery. This observation is expected, since the precision required to uncover a

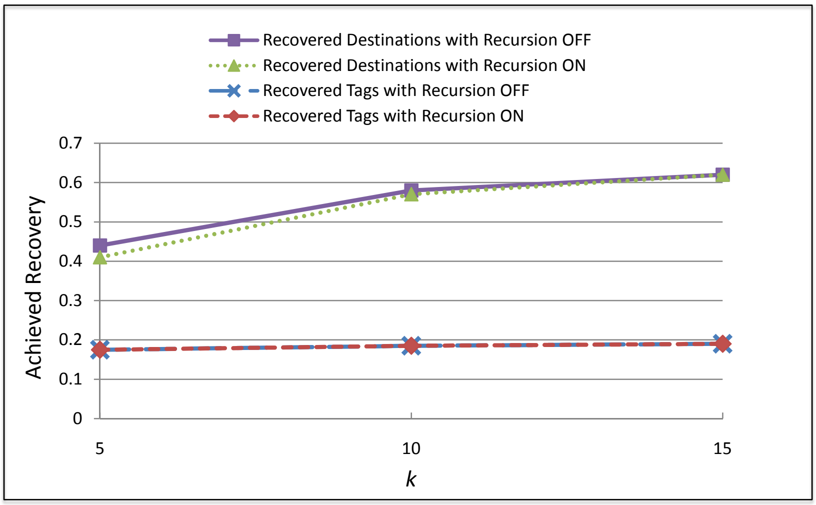


Figure 4.7: Destination and Tag Recovery for Different Values of $k \in \{5, 10, 15\}$

missing tag is much more fine-grained than that required to uncover missing destinations (i.e., the same destination u_j can be re-discovered through any of the tags in the user's profile). Second, it is worth noting that, at $k = 10$, about 60% of the missing destinations have been uncovered, when only 20% of the tags have: this illustrates how some tags are key to uncover destinations with respect to others. Finally, recursively expanding tags does not bring any gain with the dataset at hand; this is the case both when computing recovered tags and (consequently) recovered destinations. Indeed, for $k = 5$, both metrics are better off without employing recursion; this suggests that the tags added via recursion tend to broaden the actual topic of interest, hence failing to recover the intended destinations in the absence of original tags. Note that these experiments can only measure the capability of Habit in recovering tags that we *knew* were associated to the content, the same can be said for destinations, that is, we can only recover destinations that we knew were interested in the content based on the original dataset; a qualitative investigation on the expanded tagset, by means of an end-user study, would shed more light onto the relevance of those tags and destinations which have been added via expansion, without having already been present in the dataset.

4.5.2 Benchmark Analysis

In this section, we present the results of our benchmark analysis for Habit. Once again, the results are categorised according to each interest model. We first measure the performance of Habit with respect to Epidemic and Wait-for-Destinaiton protocols using people-centric interest modelling, before moving on to the analysis of Habit with respect to the distributed tag expansion technique in the context of the information-centric model. The parameters settings are those reported in Section 4.4.4, as well as those we tuned through sensitivity analysis, as presented earlier.

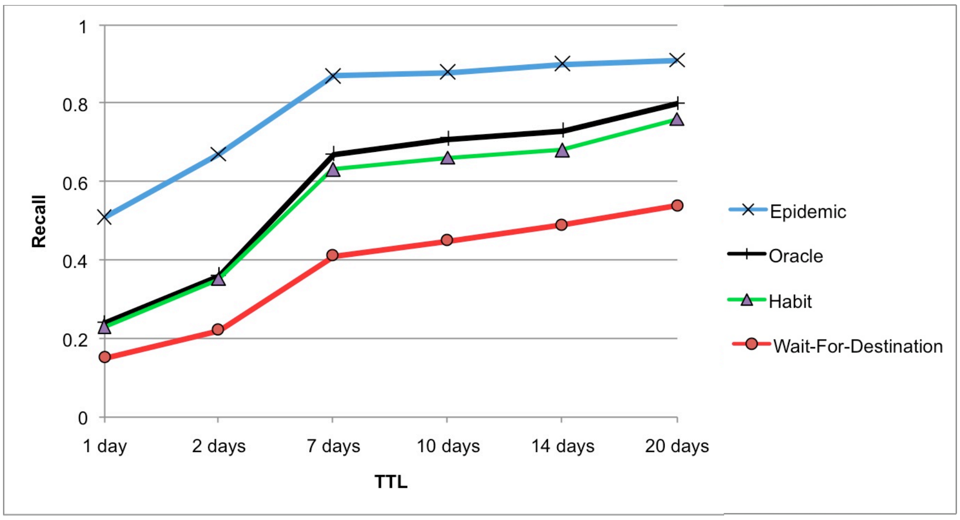


Figure 4.8: Recall [Effectiveness]

People-Centric. Let us first measure the effectiveness of Habit in comparison to our benchmark protocols. Figure 4.8 illustrates the recall, while varying the time-to-live (TTL) of the messages. As expected, Epidemic achieves the highest delivery ratio even for a short TTL, while Wait-for-Destination achieves the lowest, with a consistent gap of about 40%. As for Habit (and its Oracle variant), effectiveness is relatively low for messages with a rather short TTL; this is due to the regularity-based source routing we perform: the shorter the TTL, the lower the probability of finding a *regular* multi-hop path in the familiar stranger network connecting source and destination. However, for longer lived messages (7 days or above), the protocol achieves a level of recall above 70%, thus neatly improving over the basic Wait-for-Destination, while converging towards Epidemic. In the scenario we target, the content to be shared is likely to have long TTL (in the order of days), as it would be the case when advertising music gigs or sharing videos, as opposed to messages with a very short TTL, like traffic or weather updates. In these scenarios, people tend to meet often enough for messages to be distributed before they expire.

We then turn our attention to the main objective of Habit: efficiency. Figure 4.9 illustrates the precision measured for each protocol, again while varying the TTL. In this case, the situation is reversed: Wait-for-Destination achieves the highest precision, as messages are only passed to interested recipients upon direct encounters, while Epidemic achieves the worst, with nodes being asked to carry messages they have expressed no interest in receiving. The gap in this case is as high as 99%. The precision of the Habit (and its Oracle variant) protocol is consistently high (above 70%) across all TTL. If we combine this result with the one in Figure 4.8, in scenarios of direct interest (TTL of seven days or above), we can thus conclude that our approach is capable of achieving high efficiency (precision of 70% or above), while not compromising on effectiveness (recall of 70% or above).

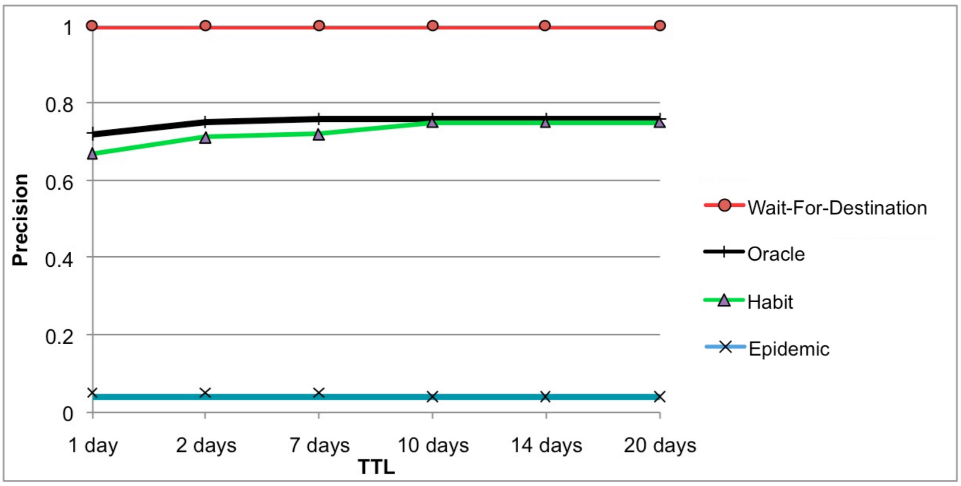


Figure 4.9: Precision [Efficiency]

We now report results in terms of the communication overhead generated by our protocol with respect to Epidemic. We have measured this overhead as the total volume of unwanted data that nodes receive throughout the simulation period. For Epidemic, unwanted data refers uniquely to the overall size of messages received by uninterested nodes; for Habit, it refers to the combination of unwanted messages plus all the metadata (regularity and interest information) that nodes exchange (up to $maxHops$ away).

We can estimate the absolute maximum size of Habit's metadata as:

$$Interest\ Information\ Size = \sum_{i=0}^{maxHops} (maxFS)^i \times I$$

$$Regularity\ Information\ Size \leq \sum_{i=0}^{maxHops} (maxFS)^i \times G$$

where I is the size of the interest profiles, and G represents the granularity of the regularity information. For instance, if nodes are logging and reasoning on hourly information, then the parameter G will be 168, referring to the number of hours in a week. However, most of the times the regularity information is smaller than the defined size, as nodes omit transmitting zero regularities. Moreover, given that people do not usually meet each other continuously every day, nodes will frequently have zero regularity values in their tables.

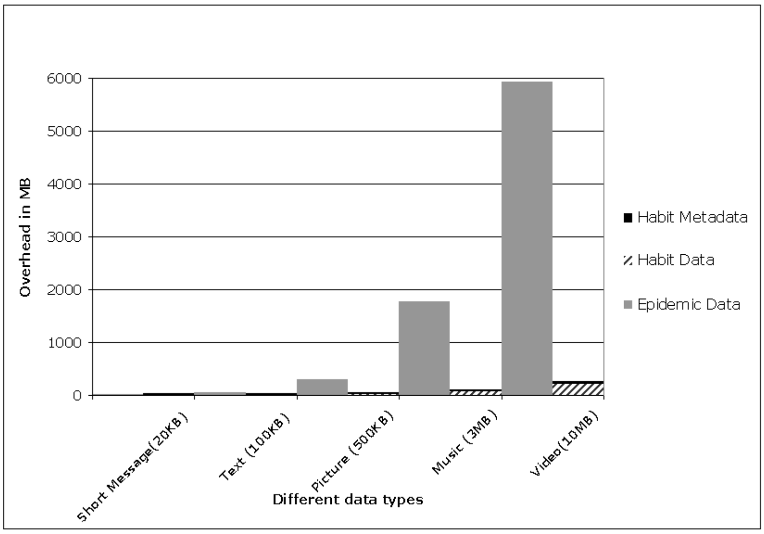


Figure 4.10: Overhead

Figure 4.10 presents the overhead, measured in megabytes, while varying the type of shared content (from short text messages to videos). For metadata, we have considered a worst-case scenario where each node exchanges complete interest network knowledge, together with regularity information of the maximum size detected during the simulation. As shown, for short and text messages of 20 KB and 100KB respectively, the amount of traffic generated by Epidemic and by Habit is comparable (that is, the flooding of content generated by Epidemic is balanced out by the dissemination of metadata by Habit). However, for media content (e.g., pictures of 500KB each, music files of 3MB each, and videos of 10MB each), the overall traffic generated by Epidemic grows disproportionally, while Habit keeps it to a minimum. This result reinforces the need for content sharing protocols that, like Habit, minimise reliance on uninterested nodes, as these are likely to cease participation in the content dissemination network if they see their resources drain while attempting to deliver messages they have not asked for. Moreover, source-based routing places the burden of computing paths on the content producer node only, while reducing the computational overhead on intermediaries to the bare minimum.

Information-Centric. We now measure the effectiveness of Habit under an information-centric model. In these experiments, we focus on the achieved recall for Habit with and without distributed tag expansion technique. For tag expansion, we have set $k = 10$ (as discussed in Section 4.5.1) and evaluated our proposed tag expansion technique, where the expanded tagset T'_c is used only for opportunistic delivery (as defined in Section 4.2.4). Figure 4.11 illustrates that tag expansion (top) increases recall to 130%, while an approach without tag expansion (bottom) has a recall of just 40%.

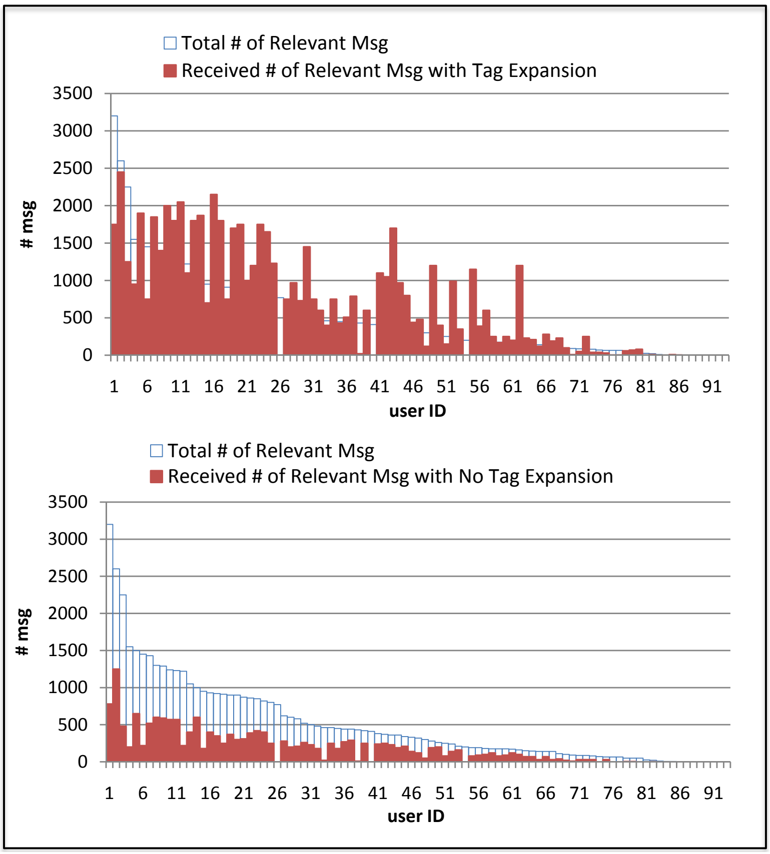


Figure 4.11: Per *Receiver's* Recall Obtained with (top) and without (bottom) Tag Expansion

In this experiment, we have measured recall (i.e., the number of relevant messages received by a node divided by the number of all messages considered relevant to it) considering as *relevant* all messages tagged with at least one tag $t_i \in P_{u_j}$ (depicted as bars with no filling in the figure). Note that this is regardless of whether the publication $[userId, tags, ts]$ referred to a movie ($movieId$) the user had actually watched and recorded in MovieLens. For example, let us assume the MovieLens dataset contains the record *[The Untouchables,* u_i*, \{classic, drama, crime, fantastic\}, ts]*; in our simulation, this would translate into a message publication *[* u_i*, \{classic, drama, crime, fantastic\}, ts]*. If user u_j has, in her profile P_{u_j}, tag *drama*, then the message would be considered of relevance to u_j, regardless of whether u_j has herself watched and tagged the movie behind the publication (i.e., regardless of whether a record *[The Untouchables,* u_j*, . . . , ts']* exists in MovieLens). As such, the increased recall merely indicates that we are able to increase the number of recipients per message, but this does not automatically guarantee that u_j is indeed interested in it. Note that the opposite also holds: the fact that u_j has not tagged a movie does not imply u_j is not interested in it, as u_j may simply not have known about it and/or not have watched it yet.

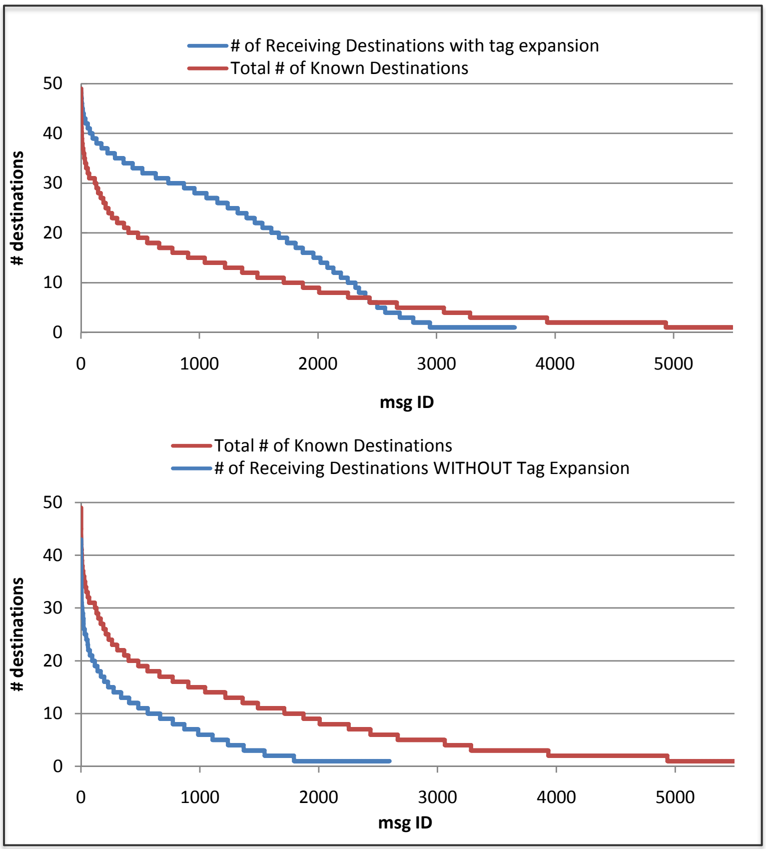


Figure 4.12: Per *Message's* Recall with (top) and without (bottom) Tag Expansion

In order to obtain a more conservative measure of recall, we have repeated the experiment but this time used MovieLens as ground truth to determine message relevance: a publication $[u_i, T_{movieId}, ts]$, stemmed from record $[movieId, u_i, T_{movieId}, ts]$, is of relevance to user u_j if and only if a record $[movieId, u_j, T_{movieId}, ts']$ exists in the dataset, this time regardless of the tags associated to it. As Figure 4.12 illustrates, when using tag expansion (top), on average 18% of deliveries that failed without tag expansion (bottom) are now able to find their destinations. However, to complete the evaluation, an end-user study should be conducted to assess precision too, in order to understand whether those extra messages being delivered by tag expansion (which do not exist in the original dataset) are indeed of interest to the end-users. We discuss this study in our concluding remarks in Section 4.6.

4.5.3 Comparative Evaluation

In this section, we report the results of a comparative evaluation against some of the major DTN protocols available in the literature. Figure 4.13 presents these results in terms of achieved recall by each protocol, while Figure 4.14 illustrates their precision. As can be seen, SocialCast outperforms all others in terms of recall. This is due to the fact that there is more importance given to the utility of colocation with the subscribers of the content (w_{col} is set to 0.75 in accordance to the original protocol), causing the messages to be constantly passed to nodes who have encountered the interested nodes more recently. However, such high performance comes with a drawback, that is, many nodes are continuously chosen to act as carriers, thus bringing the precision of SocialCast to almost as low as epidemic $\approx 5\%$ (Figure 4.14).

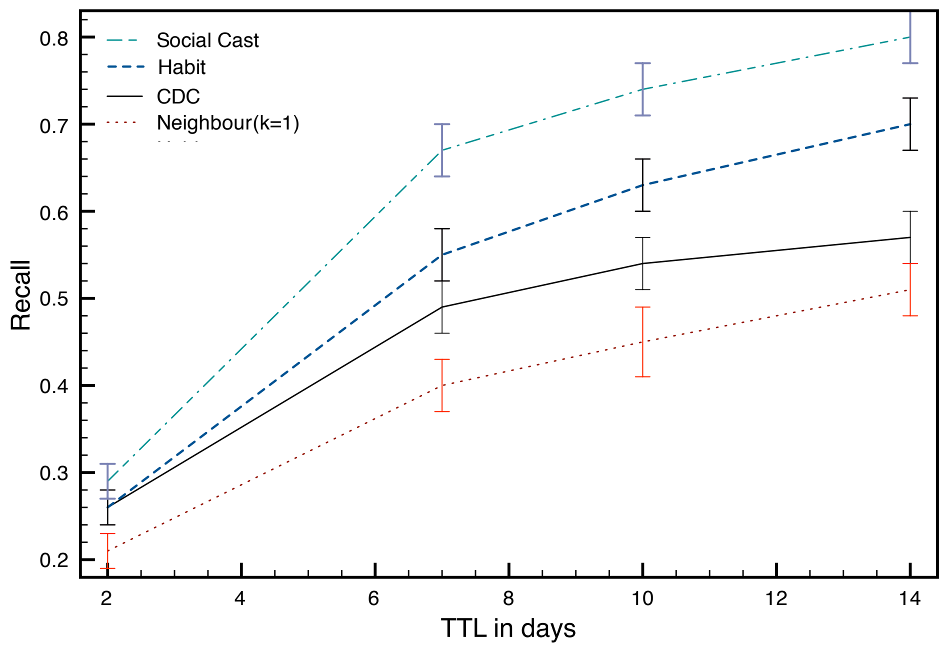


Figure 4.13: Recall for Comparative Evaluation

Neighbour(k) performs the worse amongst the other benchmark DTN protocols in terms of recall. This is due to the lack of the connectivity between communities (inter-community edges) in the friendship network as previously shown in Figure 4.4. Therefore, although the intra-community edges exist, and messages can be passed from the source to the nodes in the same friendship community, they cannot be further delivered, should the destination fall outside the friendship community. This behaviour is also observed from Figure 4.14, where no matter how big the time-to-live is, the precision stays almost the same. We also expanded our experiments with the Neighbour(k) algorithm by varying the k parameter (i.e., the maximum distance a node can be from another in the friendship network, so to be considered as carrier). However, given the property of the friendship network, where communities are very connected within but not inter-connected, the higher values of k only had little effect on performance.

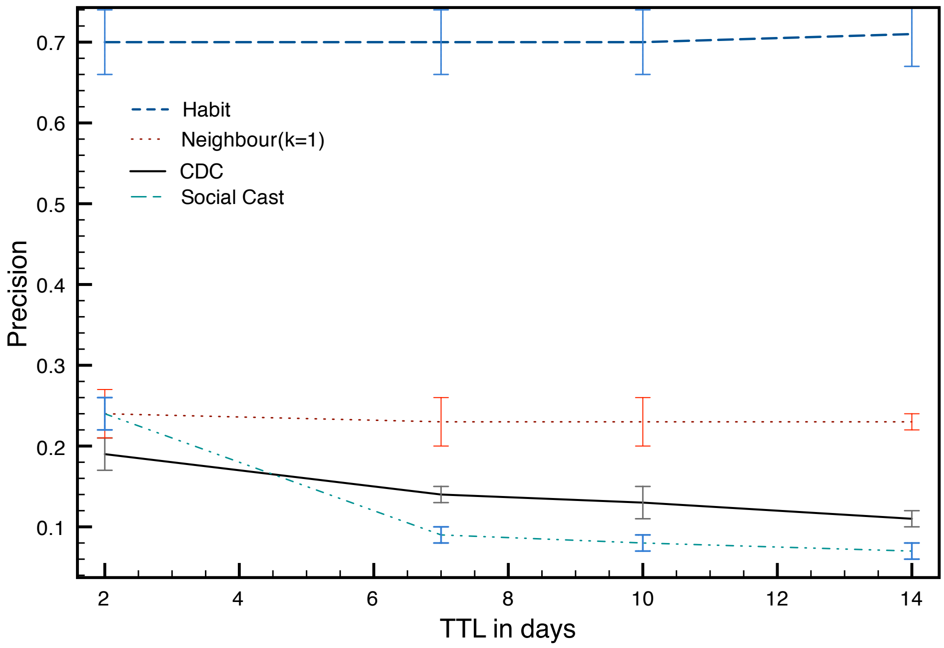


Figure 4.14: Precision for Comparative Evaluation

Finally, let us consider the performance of Habit, first in terms of recall: as the time-to-live of the messages increases, Habit achieves a higher recall; increasing the gap with CDC and Neighbour(k) by almost 10% and 20% respectively. Indeed, when messages are valid for 14 days, Habit achieves $\approx$ 70% recall, only 9% less than SocialCast. As for precision, the gap between Habit and other benchmark protocols is significant, and Habit achieves a high precision of approximately 70% for all the values of time-to-live. Habit's high and stabilised precision reflects its smart routing techniques, which unlike SocialCast and CDC avoids aggressive use of every potential carrier nodes.

In conclusion, Habit outperforms all the benchmarks in terms of precision, while still managing to have a recall very close to the leading protocols in DTN research (whose overhead though is very high). In the scenarios under consideration, Habit thus does represent the best choice.

4.5.4 Hybrid Network

In recent years, many metropolitan cities have started deploying Hot Spots, that is, nodes within a fixed infrastructure, often placed at specific locations (e.g., touristic areas, train stations, etc.) to provide a gateway to the Internet through Wi-Fi interface, often for a given service cost. Furthermore, since these infrastructures are physically fixed, they often do not have the strong resource limitations presented by mobile devices (i.e., in terms of battery). Therefore, it would be desirable, for a content dissemination network, to exploit these nodes in routing content when available, without being pre-configured to depend on them. In this regard, the question that arises is whether our proposed protocol can adaptively take advantage of any available fixed infrastructure in a hybrid network setting, where both fixed and mobile nodes exist.

To answer this question, we have evaluated Habit under a hybrid network condition modelled as follows: we assumed fixed nodes to be willing to act as relayers at all time. In so doing, we define their interest profile to include all mobile nodes in the network, as well as all the tags in the folksonomy (basically, stating that they are willing to be carriers for any messages).

In our experiments, we used information about Reality Mining data concerning cell towers availability as surrogates for Hot Spots. This dataset offers information about locations of 31545 distinct cellular towers, and has been previously used by [Sollazzo et al., 2007, Lindgren et al., 2006] to model fixed infrastructure in a hybrid network, so to assist with opportunistic delivery. Similarly, we have analysed this dataset and extracted the 10 most popular cell towers (i.e., in terms of number of distinct connections made during the 90 days of simulation) to act as Hot Spots. The mobile nodes in the network are the same as before, and have the capability of connecting to any device in range, be it a fixed node or a mobile one.

To give a flavour of the hybrid network's topological properties, we have plotted in Figure 4.15 the popularity of both mobile (ID 0 to 100) and fixed nodes (ID 100 onwards). As expected, the fixed nodes (node IDs higher than 100) are by far the most popular nodes in the network, due to their location at key areas such as the MIT Media Lab, hence meeting many nodes in the course of the day.

Based on the topological property of such hybrid network, we thus expect Habit to exploit the available Hot Spots in the same way as it exploits any other mobile node, by reasoning based on their regularities. In other words, Habit should be able to favour these fixed nodes as relayers due to their high regularity, and should do so without any prior knowledge about them or any need for the Hot Spots to behave differently.

Figure 4.16 presents recall for Habit over the described hybrid network; all parameters concerning the network and protocol have been set as in the previous experiments. As it can be

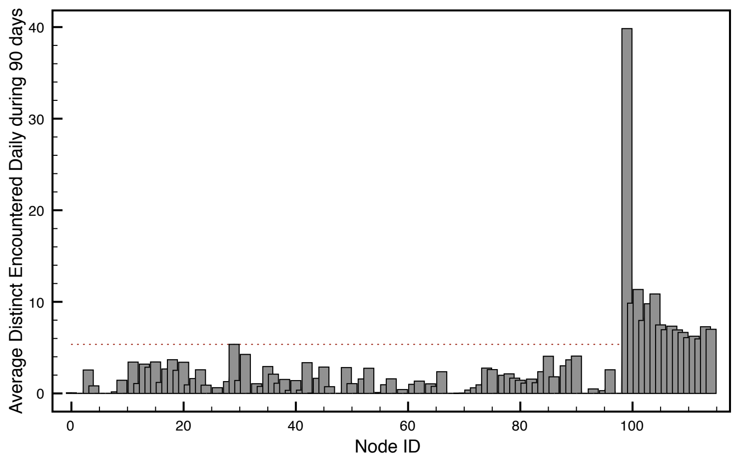


Figure 4.15: Degree of Popularity for Mobile (ID 0 to 100) and Fixed (ID 100 onwards) Nodes

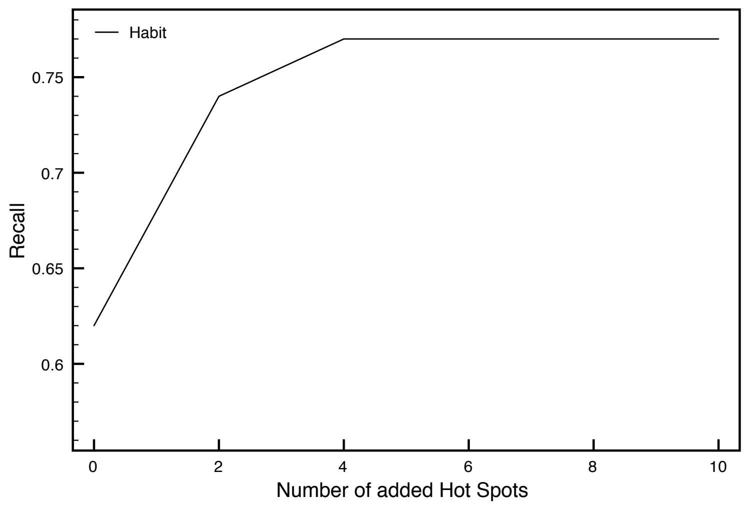


Figure 4.16: Recall for Habit over a Hybrid Network

observed, the higher the number of added Hot Spots, the higher the recall that Habit achieves, as opposed to a deployment with zero Hot Spots (i.e., Habit's original setup). Indeed, as the figure illustrates, adding 2 Hot Spots improves the performance by 12%. Another observation is that adding more than 4 Hot Spots does not bring further benefit as the achieved recall stabilises, this is because of the sparsity of the Reality Mining traces causing some nodes to be disconnected from the network (i.e., from mobile nodes and popular areas).

To address the effect of Hybrid network on message dissemination, we have also measured the number of hops messages travel through, with a hybrid network comprising 10 Hot Spots compared to a fully mobile network (infrastructure-less network). Figure 4.17 illustrates this result. We observe that, in the hybrid network, on average 13% more messages were routed by travelling through only one carrier. This is due to the previously observed topological properties of Hot Spots: more precisely, by being highly regular as well as interested in all messages, they are repeatedly chosen as carriers for many messages; once they receive a message, it is likely that they encounter destinations directly due to their high popularity (high degree of encounters with other nodes).

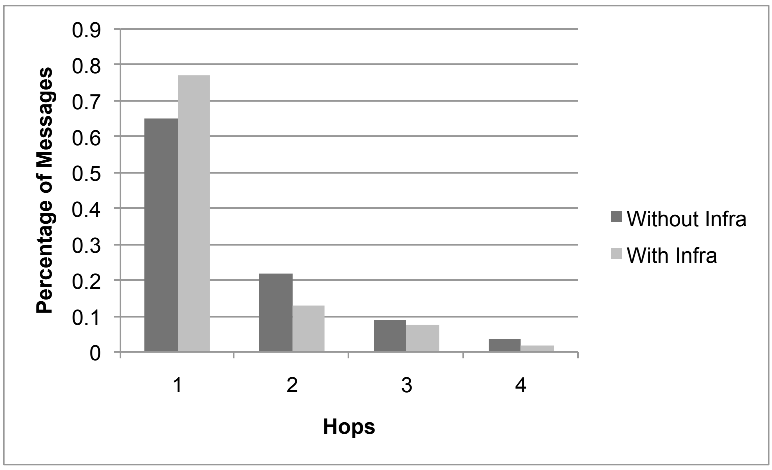


Figure 4.17: Average Hop Analysis

In this section we presented the result of evaluating Habit in a hybrid setting. In particular, we have shown that if there are Hot Spots available in the network, Habit can transparently exploit them to bring in benefits in terms of delivering more messages. Note that, using such Hot Spots and their services is often not free of charge and has a cost towards the end-user. However, as Habit has not been designed to depend on these resources and only exploits them if available, it still delivers high performance in the absence of a hybrid setting.

4.6 Conclusion

In this chapter, we have presented an interest-aware DTN routing protocol that exploits information about nodes' regularity of movement and users' interest, in order to minimise overhead, while not compromising on delivery. In so doing, we modelled user's interest in two distinct ways, and showed how interested users can be identified in the absence of a centralised authority. We presented the very first distributed tag expansion technique for mobile networks, and showed that many previously un-identified destinations can be uncovered. However, it remains open to evaluate the *precision* of our proposed technique, by means of a comprehensive user study, so to allow us to assess whether the suggested destinations (users) were indeed interested in the content.

We evaluated the realisation of our proposed content dissemination protocol (Habit), by means of simulation on real traces, and showed that its source-based routing is well suited in scenarios where high volumes of media content are being generated and shared via resource constrained devices. Furthermore, we performed a thorough comparative evaluation of performance that Habit offers, with respect to state-of-the-art protocols, accounting for both precision and recall. Finally, we evaluated Habit over a hybrid network and showed that it can gain in performance by dynamically adapting to the environment and exploiting infrastructure nodes, should they become available.

In this chapter, we have looked at the problem of user's participation from the point of view of user's interest. However, there are other factors impacting user's participation. In the next chapter, we thus fix the dimension of user's participation based on interest, by focusing on a people-centric approach, and turn our attention to participation as dictated by resource constraints. In particular, we focus on battery constraints on mobile devices and introduce a load-aware approach for distributing the workload fairly amongst relayers.

Chapter 5

Load-Balancing

User's participation in a content dissemination network is not only driven by their own interest, it is also dictated by available battery and the (selfish) need to preserve it.

In this chapter, we investigate the problem of user's participation often faced with limited *energy* available (Ch 3 as described in Chapter 1). Indeed, research has shown that battery constraints on mobile phones are not expected to be significantly lifted in the upcoming years [Paradiso and Starner, 2005] (i.e., Moore's Law will apply to the miniaturisation of battery size, rather than increasing its lifetime), making energy efficiency an important topic in mobile networks. New protocols [Wang et al., 2007, Jun et al., 2006] have been proposed with the aim of saving up energy, by limiting the number of replica messages in the network while not compromising delivery. However, reducing the overall network overhead does not imply this is done *fairly* across all nodes involved in the content distribution network. Indeed, these approaches reduce the *overall* network overhead by exploiting users mobility patterns and/or by reasoning on network topology; as our research will demonstrate, this leads to a small subset of nodes to be repeatedly selected as content carriers over and over again, leading to highly unfair workload distribution. As these nodes will inevitably see their battery drain very quickly, they are more likely to cease participating in the content delivery network, with detrimental effects on the overall delivery.

In this chapter, we introduce our load-balancing mechanism for participatory DTN that, once integrated with source-based routing protocols, achieves *fair workload distribution over time* without compromising delivery. The rest of this chapter is structured as follows: first we motivate this work by presenting the results of a comparative evaluation of some of the state-of-the-art DTN protocols, demonstrating how they *all* cause heavily unfair workload distributions; we then review the literature in the broader area of load-balancing, from which we draw inspiration for this thesis contribution (Section 5.1). In Section 5.2 we present our approach: we begin with a detailed description of the model and its two key components, that is, load prevention and load alleviation, and continue discussing a specific implementation, CoHabit, in Section 5.3. We then proceed to evaluating CoHabit by reporting our simulation settings in Section 5.4, and demonstrating the results of our evaluation, by illustrating how fairness can indeed be achieved without compromising delivery in Section 5.5. We then conclude this chapter, summarising our achievement and contributions in Section 5.6.

5.1 Background

5.1.1 Motivation

In DTNs, the success of routing protocols heavily relies on the participation of nodes in the network. However, participation cannot be taken for granted: mobile devices have a rather limited amount of battery; if a device is asked to forward many messages in a brief period of time, it will deplete its battery at unexpected speed, causing its user to most likely cease participation in the content delivery network altogether.

To investigate the magnitude of the problem, we have conducted the following analysis: we have taken real mobility traces of two typical DTN settings: Reality Mining traces corresponding to 96 staff and students (as was described in Section 4.4.1), moving around the MIT university campus, for a period of nine months [Eagle and Pentland, 2006]; and 100 cabs, moving within the San Francisco bay area, for a period of one month [Piorkowski et al., 2009]. In both cases, nodes were posting messages to a varying number of recipients and at a varying publication rate according to a real dataset as we will further discuss in Section 5.4. Two different DTN's latest generation protocols were used to distribute these messages: *Habit* [Mashhadi et al., 2009] presented in Chapter 4, a source-based routing protocol which leverages both social and mobility networks to select delivery paths; and *Change in Degree of Connectivity* [Musolesi and Mascolo, 2008] (*CDC* as previously discussed in Chapter 2), a mobility-based protocol that selects as carriers those nodes who exhibit higher popularity (i.e., who encounter the largest number of distinct nodes in a given period of time). In particular, we have chosen Habit to assess how fair a protocol that already reasons on user interest and mobility is in spreading the load in the network. We have chosen CDC as it represents an effective (i.e., high delivery) mobility-based protocol with low network overhead.

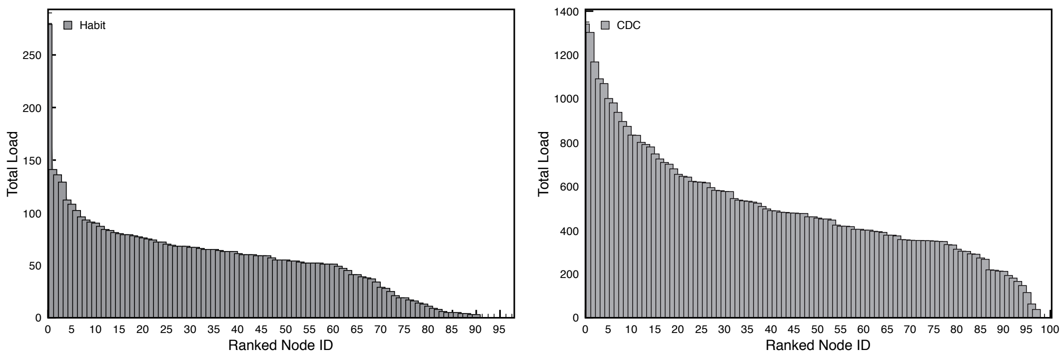
In this experimental setup, the maximum battery lifetime of a device has been modelled as the number of messages the device can *forward* in a given time period; we call this limit the *drainage threshold*. If this threshold is reached, the device ceases participation altogether, that is, it stops forwarding messages for others. Note, however, that we still allow devices to send the messages they produce, as well as to receive the messages they are interested in (selfish behaviour). In so doing, we abstract devices' participation in the content delivery network as the number of messages they forward within a time period; this provides a simple yet realistic indication of energy consumption (communication has been measured to be the highest energy draining factor in mobile devices [Miluzzo et al., 2008]). To give a flavour of how much actual data can be transferred for different drainage thresholds, we have performed the following calculation: we have considered the HTC Fuze smartphone, whose overall battery capacity is 1340mAh, and assumed that, in a period of 5 days, people are willing to recharge their phone at most twice (thus leading to an overall capacity of 3 * 1340mAh = 4020mAh); we have then assumed that, of this capacity, users are willing to devote at most 50% (i.e., 2010mAh) to opportunistic content forwarding. Based on the energy consumption study presented in [Balasubramanian et al., 2009] for data transfers using Wi-Fi on the same HTC Fuze smartphone, we have then calculated the maximum number and size of messages that a node can send over a period of 5 days as presented in Table 5.1.

Number of Messages	Message Size (Type)
300 / 5 days	800KB (Text)
150 / 5 days	1.6M (Picture)
75 / 5 days	3.2M (Music)
35 / 5 days	6.4M (Video)

Table 5.1: HTC Fuze Smart Phone (Light) Consumption Measurements

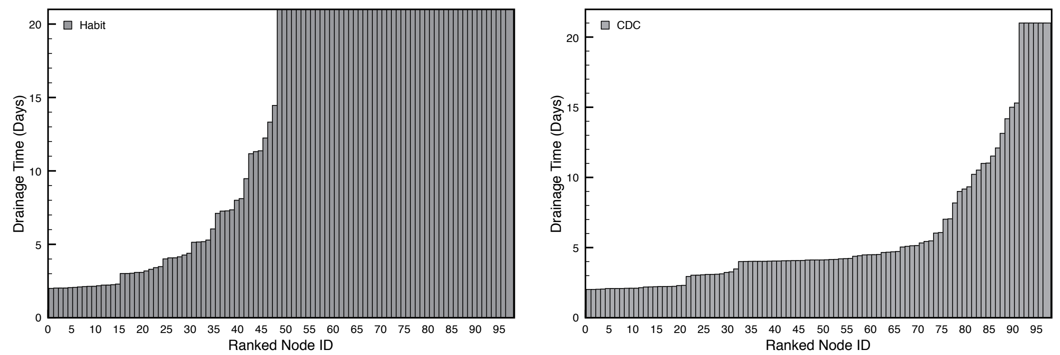
Note that the reported values are quite optimistic, and only representative of scenarios where users make a rather light use of their mobile phone (e.g., for texting and calling, so that they only need to recharge every 2 days), and are thus willing to devote up to 50% of their battery to content dissemination. However, users who make a more advanced use of their device (e.g., for browsing, social networking, emails, etc.), are unlikely to devote that much battery to DTN content sharing, as they already have to recharge their phone once per day just to support their normal use [CNET, 2010]. The number of messages that these users are willing to forward is likely to be less than what reported in Table 5.1, thus making the workload distribution problem even more severe.

In our analysis, we gave each node the same drainage threshold, and assumed all messages count the same in terms of consumption (e.g., they are media files of comparable size/type). Using this setup, we have measured the workload distribution among nodes in the network. Figure 5.1a and 5.1b illustrate the results for Habit and CDC respectively, when considering



(a) Habit with Threshold $\frac{50msgs}{5days}$ (b) CDC with Threshold $\frac{100msgs}{5days}$

Figure 5.1: Load Distribution



(a) Habit with Threshold $\frac{50msgs}{5days}$ (b) CDC with Threshold $\frac{100msgs}{5days}$

Figure 5.2: Life Expectancy

the cabs traces. Similar results were obtained when considering the Reality Mining traces. As shown, the load-per-node varies considerably in both cases, with a few nodes performing significant amounts of work, while others contributing very little. While it can be expected that those nodes sitting at the very edges of the network will always contribute less to the overall content delivery, a fairer load distribution should be attained among all others, in particular, relieving the most central nodes from part of their load. Note that, while in Habit a threshold of 50 messages in a 5-day period was used, for CDC we used a much higher one (i.e., 100 messages per 5-day); this is because CDC makes an aggressive use of colocated devices to deliver content (i.e., via multiple message replicas), causing the whole network to reach the (lower) threshold very rapidly, thus ceasing participation immediately.

As the above experiment confirms, DTN protocols that reason on node mobility patterns to disseminate content, distribute the load very unfairly. This would not be a significant problem per se, if the load were distributed over a long period of time, so that the drainage threshold was rarely reached. Unfortunately, this is not the case. We have conducted a second experiment where, under the same simulation setup, we have measured the number of days nodes continue distributing content, before the drainage threshold is reached. We assume that, if nodes see their battery drop suddenly, they cease participation, no matter at what point in time this occurs. The results are shown in Figure 5.2a and 5.2b for Habit and CDC respectively (once again for the cab traces). As shown, there is a significant number of nodes that cease participation in the very first day of the simulation, as they are immediately selected as content carriers for many messages; by the end of the simulation period, 50% of nodes using Habit have reached the drainage threshold and left, and the percentage becomes as high as 90% for more aggressive protocols like CDC. This has immediate and detrimental impact on content delivery. It is thus clear that, for DTN protocols to become a viable means to share content, they must be designed from the outset to spread load more fairly, while not compromising delivery.

5.1.2 Related Work

The problem of unfair load distribution in networked systems has been the subject of extensive research, from Internet-based settings (e.g., [Kleinberg et al., 2001, Cao et al., 2002, Rao et al., 2003]), to more challenged environments, such as mobile ad hoc networks (e.g., [Li and Cuthbert, 2004, Marina and Das, 2002, Zhang et al., 2002, Yin and Lin, 2005, Hassanein and Zhou, 2003]). In MANETs, the high density of network connections, coupled with their relative stability, made it possible for routing protocols to use a feedback mechanism in order to reason upon, and equally spread, the network workload. For example, the approach presented in [Zhang et al., 2002] relies on probing packets, which are sent periodically to each different disjoint path toward a given destination; based on their round-trip time (RTT), the path delay is estimated and the congested nodes identified and avoided in the future. Similarly, in MALB [Yin and Lin, 2005], each source sends a probe message to its destinations; nodes receiving the probe reply, also stating their battery level, thus enabling the source to estimate both congestion and available power resources along each path. However, while successful in a MANET setting, these approaches cannot be directly applied in DTN; in fact, they rely on both the probes and the feedback packets to be delivered quickly, so that the information they carry about the network is fresh and up-to-date. In DTN, however, this is usually not possible, as the network is much sparser, and nodes have a much longer inter-contact time, making probe and feedback mechanisms not suitable.

In the context of DTN, there has been a little work concerning fair distribution of workload and its implication on energy-constrained mobile devices. In [Solis et al., 2010], the problem of fair resource allocation, in particular concerning *buffer management*, is addressed in networks where a percentage of users refuse the contribution that is required to drive the content dissemination network, while using the common pool of network resources. The authors show by means of simulation the effect of such *malicious* behaviour in reducing the portion of successfully delivered messages of the *honest* (i.e., non-malicious) users. To tackle this problem, a technique based on prioritising messages by relying on authenticating the users is introduced. [Boldrini et al., 2008] similarly focuses on buffer management and addresses the problem of the trade-off between the cost associated to receiving a piece of content in terms of resource consumption and the value of the content to the end-user. Their approach reasons on the value that content is worth not only to the end-user but to the community that the user has acquaintance with, while also taking into account the cost of storing the content in the node's *local buffer*. In [Ye et al., 2009] and [Guo and Keshav, 2007], approaches concerned with resource allocation in message ferrying for rural villages are introduced. [Ye et al., 2009] ensures a fair service amongst the served villages while minimising the transit delay. This is done by reasoning on optimisation of the overall system performance, such that the limited message capacity of the ferries is allocated to messages that minimise transit delay, while avoiding message starvation for afar villages. In [Guo and Keshav, 2007], the problem of fair bandwidth consumption amongst villages has been addressed by introducing a scheduling algorithm that aims to minimise the network transit delay while achieving fairness. However, these approaches are particularly introduced to respond to the needs of rural scenarios, where the focus is on delay and limited storage.

In RAPID [Balasubramanian et al., 2007], a utility-based approach is introduced which can optimize delay-related metrics (e.g., average delay, missed deadline), by treating routing as a resource allocation problem. The uniqueness of the protocol lies in taking constraints of both bandwidth and storage into account, in order to determine how packets should be replicated in the system. ORWAR [Sandulescu and Nadjm-Tehrani, 2008] is another approach that also reasons in terms of utility, but focusing on network-wide optimization, in order to minimise the probability of partially transmitted messages. [Radenkovic and Grundy, 2010, Seligman et al., 2007, Grundy and Radenkovic, 2010] focus on the issue of congestion in DTNs, tackling the problem of storage exhaustion. Orthogonal to this work, in [Radenkovic and Grundy, 2010], the authors focus on forwarding algorithms that adaptively select the next hop carrier based on users mobility and their encounter history, and propose a new forwarding approach which reasons on delay and buffer capacity. Although all of the above approaches focus on the problem of resource allocation in similar settings as the scenario in this thesis, they mainly focus on storage and/or bandwidth allocation.

The only other attempt to achieve fairness in DTN is FairRoute [Pujol et al., 2009], a routing protocol that uses concepts from social sciences (e.g., interaction strength, social status) to select message carriers. We briefly describe FairRoute next, and throughout this chapter refer to it as a comparison benchmark.

FairRoute [Pujol et al., 2009]. FairRoute takes its inspiration from SimBet [Daly and Haahr, 2007], a routing protocol which we described earlier in Chapter 2, that chooses intermediaries based on betweenness centrality and probability of future interactions. Similarly, FairRoute relies on social interaction strength in human society, in order to select paths from source to any destinations. This interaction strength between any two nodes in the network represents the likelihood of the contact to be maintained over time. Such social interaction strength is then defined in terms of short and long strengths, which indicate relations in short and long time-scale respectively. Depending to the mobility of users, a long inter-contact time between two nodes results in both short and long interaction strengths to be decayed by a decreasing factor (i.e., this value is trace-driven and requires to be known beforehand). Upon an encounter, the next hop relayers are chosen based on an *aggregated interaction strength* between the encountered node and the destination.

In order to ensure a fair workload distribution, FairRoute relies on assortative behaviour of social interactions amongst individuals. This behavioural property is described in the context of human society as selection of those we choose to spend our time with, thus interacting with people of the same class and tending to disregard interactions with individuals from lower social status. FairRoute captures this property in order for nodes to share their resources, by limiting interactions such that an intermediary would only accept messages from nodes who belong to a higher class. The social classes are then defined in terms of popularity of nodes, reflected by their queue (i.e., buffer) size. In time, this causes central heavy loaded nodes to offload their load to less loaded nodes.

The authors demonstrate, by means of simulation on the Reality Mining mobility traces, that their protocol achieves more fairness than protocols such as Epidemic [Vahdat and Becker, 2000] and SimBet [Daly and Haahr, 2007]. However, one important dimension that FairRoute neglects is *time*: while it distributes load fairly over the *simulation time as a whole*, battery may still drain very rapidly when short periods of intense connectivity follow long periods of disconnection, with direct impact on delivery, as well as long-term node participation. Indeed, our experience with the very same dataset has shown that load comes in very rapid bursts, because of the high inter-contact time between nodes. We will demonstrate this observation in Section 5.5.

In the next section, we present the first load-balancing mechanism that achieves fair content dissemination continuously over time, *regardless* of the inter-contact time distribution between nodes in the network.

5.2 Conceptual Model

Our approach to content dissemination in DTNs aims to distribute load fairly, while still achieving a high delivery ratio. In this section, we present the load-balancing model proposed to achieve this goal, while the next section illustrates a specific realisation of the model within a DTN source-based routing protocol.

Our load-balancing model consists of two parts: *Load Prevention*, whereby each node locally estimates other nodes' workload, in order to source-select paths that rely on the less-loaded part of the network; and *Load Alleviation*, whereby each node monitors the traffic it is asked to forward, reacting to bursts by means of a controlled forwarding mechanism that pre-emptively reduces participation, without reaching drainage. We describe each component next.

5.2.1 Load Prevention

In order to distribute load on nodes fairly, two challenges need to be addressed: first, the workload of nodes must be locally estimated, without relying on end-to-end shared knowledge; second, based on this knowledge, message routes must be chosen so to effectively spread the load fairly, whilst not affecting delivery.

Challenge 1 - Load Detection. Creating accurate estimates of nodes' load is a challenging task in a DTN setting: in fact, load varies dynamically over time, and if one has to use such information to prevent overloading, only very up-to-date information is of value. Disseminating load information in a DTN setting thus makes little sense, as such information does not tolerate delay.

In source-based DTN, however, there is a great deal of information already encoded in the headers of the messages that can be locally exploited to estimate the network workload. Each node u_i locally maintains an *Estimation Log*, which logically consists of a table with one row for each known node in the network, and one column for each *slot* in a given *observation period* (e.g., with a slot of one day over an observation period of five days, there would be five columns). Each cell in the log records how many times a known node u_j has been used as carrier in that slot (e.g., on that day). The counter is updated in two circumstances: (1) node u_i creates a new message and computes source-based paths for its delivery, containing node u_j at some point as intermediary; (2) a message is routed through u_i, containing u_j in its header as another intermediary somewhere along the path. Note that the Estimation Log is not time-accurate: in fact, although the log is increased when a message is either created on u_i or when it reaches u_i, the intermediary u_j could indeed be affected by this load at another point in time. Nonetheless, given that it is not possible to know exactly when the message will be routed through u_j, this technique provides valuable information to be used during load distribution. Moreover, it is not the load recorded in a single slot, but an *aggregated* load value, computed over the whole observation period, that is used, as described next.

Challenge 2 - Load Distribution. Source-based routing protocols enable nodes to retain control over the routes followed by the messages they produce. Such property is very attractive as we can leverage it to build a load-aware reasoning scheme, whereby sources inspect the estimated load of intermediaries, and consequently select paths that distribute load more evenly. More precisely, we associate to each path an estimate of its *current overall load*. This value, computed by consulting the Estimation Log, is the sum of the load values recorded for each intermediary node in the path, and for the past observation window ΔT. A path is then *probabilistically selected* based on its expected utility:

$$Util_{path} = 1 - Load_{path} \tag{5.1}$$

where $Load_{path}$ is a value in the range $[0, 1]$, representing the current load of the path, normalized over the most loaded path found. Once a path has been selected, the source node logs its intermediaries in the Estimation Log.

5.2.2 Load Alleviation

We claim that, no matter how good the load prevention technique is, nodes can still become overloaded if the inter-contact time between nodes is long, and the system is under a high rate of message publications. In such circumstances, nodes may end up storing many messages for various destinations which are less frequently encountered (i.e., high inter-contact time); if such nodes are then encountered in a relatively short time period, then an enormous amount of traffic is generated in a very short timeframe, causing local battery to rapidly exhaust (i.e., the node's actual load will hit the drainage threshold).

The load alleviation component has been developed to monitor nodes' actual load, enabling them to step back from participation in relaying for others when a critical load threshold is reached, and giving them enough time to recover before resuming participation. Each node locally keeps track of the actual number of messages it has forwarded for each slot (e.g., each day) within a given observation period (e.g., $\Delta T = 5$ days); if the total number of messages forwarded within an observation period reaches a certain *Critical Boundary*, the node temporarily stops relaying messages, until it is safely out of the critical zone. The Critical Boundary is defined as a percentage of the Drainage Threshold; for example, if the Drainage Threshold is 100 messages within 5 days, and the Critical Boundary is 90%, then the load alleviation component restrains a node from further message forwarding once *Critical Boundary* $\times$ *Drainage Threshold* $= 90$ messages have been sent in the past $\Delta T = 5$ days. Due to the sliding of the ΔT observation window, the refuse-to-relay is only temporary, and once enough time has passed (i.e., once their load during the last ΔT has fallen under the Critical Boundary), they resume participating in the content delivery protocol.

5.3 Realisation

In order to evaluate the fairness that our load-balancing model brings, we required an underlying source-based routing protocol. In this regard, we have selected Habit, the source-based routing protocol we previously described in Chapter 4. As demonstrated in Section 5.1.1, Habit fails to distribute load fairly, hence causing some nodes to cease participation. We have thus implemented our load-balancing model on top of Habit, and we refer to this new protocol as CoHabit. To reconcile Habit's goal of minimising the number of uninterested relayers with fairness, the load prevention component is simply modified so that a path is probabilistically selected based on its expected utility, which is now a combination of load and cost (as per original Habit):

$$Util_{path} = 1 - \left(\frac{(1-\alpha) \cdot Cost_{path} + \alpha \cdot Load_{path}}{2} \right), \tag{5.2}$$

where *cost* is quantified in Habit as the number of uninterested intermediaries in a path, normalized over the most costly path found, so to vary in $[0, 1]$, as previously described in Section 4.2, $\alpha \in [0, 1]$ is a weight that can be tuned to give more importance to load or cost. To provide more details on this specific realisation, we next present CoHabit's pseudo code, where the highlighted lines represent changes with regard to Habit.

$Parameters$

- s: source node, R: set of recipients
- $Paths$: the set of paths from s to all $r \in R$
- $Neigh[u]$: Node u's direct neighbours in the regularity graph
- $Reg(u, v, t_{now}, t_{exp})$: first non zero regularity weight w_{reg} between u and v occurring at

time $t \in [t_{now}, t_{exp}]$; or -1 otherwise

- $LoadEstimation[u]$: aggregated load estimation of node u as seen by the source s
- MAX_LOAD: heaviest loaded path among those found
- MAX_COST: most expensive path among those found

$FindPaths(s, R, t_{now}, t_{exp})$ {
$Paths = \emptyset$;
for all $r \in R$ {
    $current.path = \emptyset, current.cost = -1,$
    $current.load = 0$
    $RecursePaths(s, r, current, R)$
    }
$SelectPaths(R)$
}

$RecursePaths(u, r, current, R)$ {
if $(u \notin current.path)$ {
    $current.path = current.path \cup \{u\}$
    if (u==r) {
        $Paths = Paths \cup \{current\}$
        return
        }
    if $(u \notin R)$ $current.cost++$;
    $current.load = current.load + LoadEstimation[u]$
    for all $v \in Neigh[u]$ {
        w_{reg}=$Reg(u, v, t_{now}, t_{exp})$
        if $(w_{reg} > 0)$
            $RecursePaths(v, r, current, R)$
        }
    }
}

$SelectPaths(R)$ {
for all $r \in R$ {

$maxUtil = 0, selectedPath = \emptyset, SelectedPaths = \emptyset$

for all $p \in Paths \mid p.destination = r$ {

  $load_p = p.load/MAX\_LOAD$

  $cost_p = p.cost/MAX\_COST$

  $util_p = 1 - \left(\frac{(1-\alpha)\cdot cost_p + \alpha \cdot load_p}{2}\right)$

  if $(maxUtil < util_p)$

    $maxUtil = util_p$, $selectedPath = p$

  }

$SelectedPaths = SelectedPaths \cup \{selectedPath\}$

}

}

In the next section, we evaluate CoHabit and demonstrate that, while Habit achieves high delivery by means of path choices that are often unfair (i.e., that results in the same nodes always being selected as intermediaries between a given source and destination), CoHabit maintains high delivery while also balancing the load evenly. We first present our simulation settings in Section 5.4, before discussing the result of our extensive evaluations in Section 5.5.

5.4 Simulation Settings

In this section, we define our simulation environment, carefully accounting for scenarios where network is heavily loaded. In order to evaluate our proposed load-balancing technique, we used the OMNeT++ [OMNeT++, 2010] event simulator to measure performance of CoHabit and other benchmark protocols as we shall define next.

5.4.1 Datasets

To evaluate CoHabit under a realistic setting, we required three distinct sets of information: human mobility traces (to simulate encounters), users' social network (to determine who is interested in receiving content from whom), and publication rate (to simulate how often new content is generated). To date, there is no available dataset offering all this information at once; we thus selected three distinct datasets, each providing one piece of information, before overlaying them together as we will describe later. The selected datasets are the following:

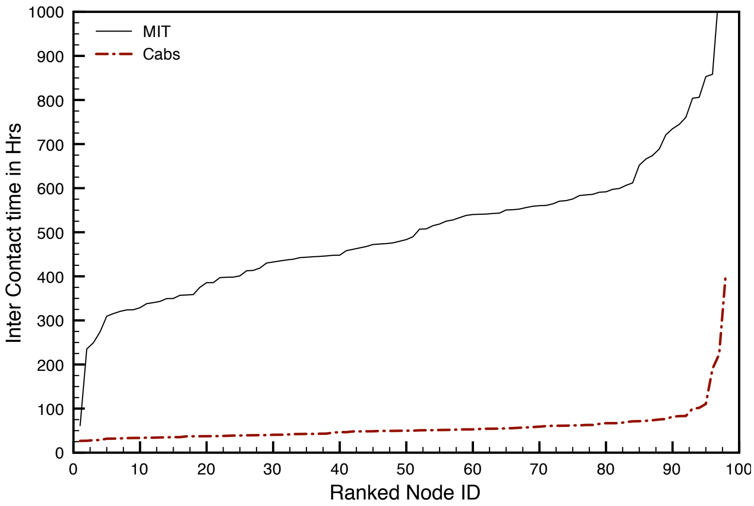
Mobility Traces: in terms of modelling user mobility, we experimented with two real mobility traces of different topological properties: the Reality Mining traces, containing mobility information of staff and students at the MIT campus (as we previously described in details in Section 4.4.1), and a vehicular dataset of cabs in San Francisco Bay area

[Piorkowski et al., 2009].

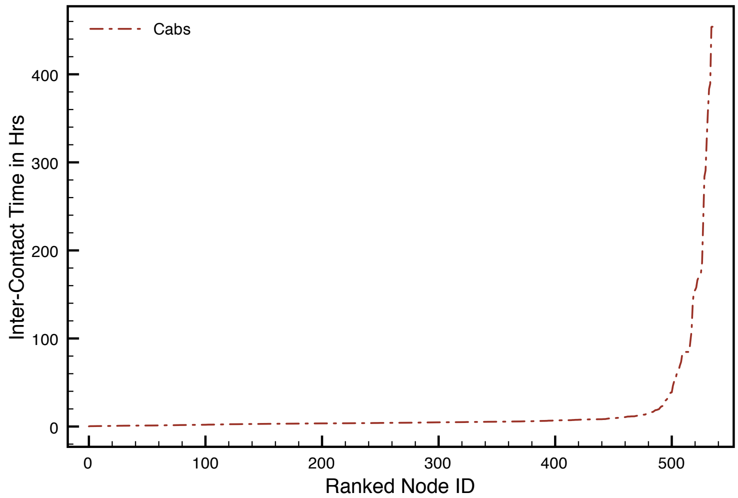
Reality Mining Traces: for these experiments, we extracted five months of colocation data, from September to February; we used the first five weeks of these traces as training period for Habit to discover nodes regularity of movement (as described in Section 4.5.1); the remaining period was then used as the actual test period for CoHabit, with nodes creating and sharing content. The Reality Mining dataset has been widely used to evaluate DTN protocols [Zhang, 2006]; while being representative of some DTN settings (i.e., university campus), its sparsity and very high inter-contact time is not representative of DTN urban settings, where nodes are much more frequently connected (as in a MANET), but with short contact time (as in a DTN). To cover this scenario, we have thus expanded our evaluation to include an urban vehicular dataset, described next.

San Francisco Cab Traces: these traces recorded the GPS coordinates of 500 cabs, logged every 10 seconds, over a period of 21 days, in the San Francisco Bay Area. In order to infer colocation information from GPS coordinates, we have assumed that two cabs are colocated if their physical distance is less than 50 meters (i.e., within Wi-Fi range); furthermore, as cab clocks are not synchronized (i.e., two cabs may be physically colocated, but may log their respective location with a few seconds difference), we have assumed a 60 second interval during which, if the distance between two cabs is less then 50 meters, those cabs are assumed to be colocated. As the period covered by these traces is quite short, we have replayed the traces twice back-to-back, with the first 21 days used as training period by Habit, and the subsequent 21 days used to actually test CoHabit (i.e., with nodes creating and distributing content). It is worth emphasising that, at this point, we are not evaluating Habit and its ability to learn regularity, so that this re-playing of traces does not distort our evaluation of CoHabit.

Furthermore, for the purpose of our experiments, we have sampled a subset of 100 randomly chosen cabs while preserving the characteristics of the original dataset such as the distribution of inter-contact time. Figure 5.3a and 5.3b depict the inter-contact time for the sample and the full dataset respectively, demonstrating that the distribution is preserved. Moreover, Figure 5.3a draws a comparison between the two described mobility traces Reality Mining and Cabs, highlighting their topological differences. As it can be observed, the inter-contact times for cabs is very short



(a) Average Inter-Contact Time for Sampled Cabs and Reality Mining

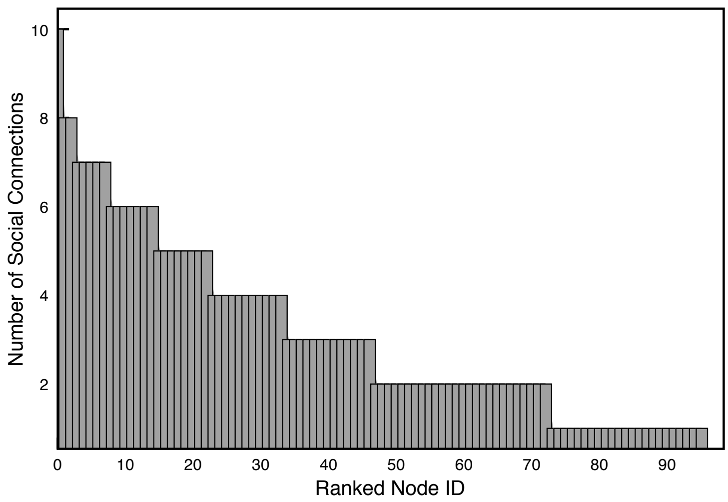


(b) Average Inter-Contact Time for the Full Population of Cabs Traces

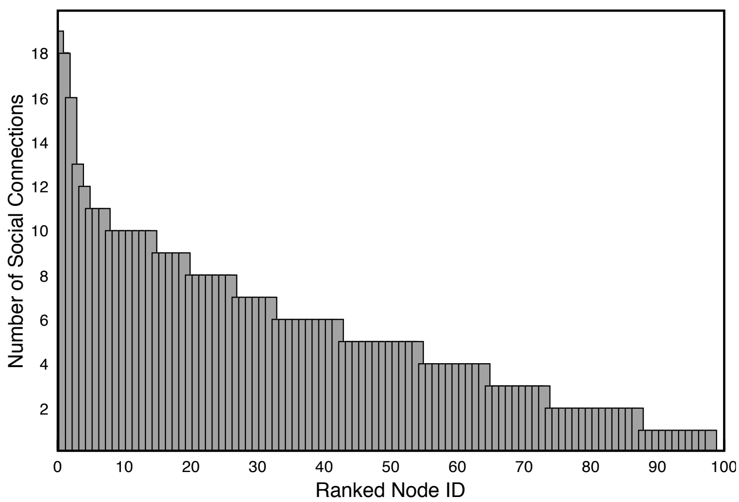
Figure 5.3: Topological Properties of Different Mobility Traces

and in order of 50 hours. On the contrary, the Reality Mining mobility traces are very sparse and exhibit very high inter-contact time (in the order of 500 hours).

Interest Network: as discussed earlier in Chapter 4, user interest can be modelled in two ways: information-centric and people centric. In order to concentrate on the load-balancing aspect of our protocol, we have confined our experiments to the people-centric model, without loss of generality as the very same reasoning applies to the information-centric model too. We have then experimented with two social networks to determine who is interested in receiving content from whom. The first is based on implicit extraction of



(a) Reality Mining



(b) Last.fm

Figure 5.4: In-degree Distribution of Selected Social Datasets

exchanged text and voice messages from the Reality Mining dataset (as was previously described and used in Section 4.4.1). Figure 5.4a shows the in-degree distribution for this interest network, describing a long-tailed distribution, where a few users are interested in many sources, while the vast majority is interested in receiving content from a few source only. As the inferred interest network already complies with the Reality Mining mobility traces, we do not require mapping the two. However, in order to overlay an interest network on the cab traces, we required another social network. To further assist us in modelling a heavily loaded network, we selected a denser social network. In particular, we have taken a sample of the self-reported social network of Last.fm

[Last.fm, 2010], a music social networking website. Last.fm is primarily a music recommendation service, which provides alternative tracks to users by monitoring the songs they play. Furthermore, as any other social networking website, it allows users to specify their social network by adding contacts. Last.fm provides a service called "Audioscrobbler" [Audioscrobbler, 2010] through which it is possible to extract user IDs and their list of contacts. To sample this dataset, we have first gathered 10,000 Last.fm users with a breadth-first search using the Audioscrobbler Web Services; we have then sampled a (connected) sub graph of 100 users; an analysis of their in-degree distribution highlights the long-tailed degree distribution typical of human social networks as illustrated in Figure 5.4b. In particular, this sampled dataset exhibits a more connected social network in comparison to the Reality Mining social network (Figure 5.4a), with the mean number of 5 connections as opposed to only 2 connections in the Reality Mining social network.

Publication Dataset: finally, each user in our simulation is asked to create and inject content in the network. To evaluate our approach under a heavy load of messages, we require users to publish with an accordingly high publication rate. In real settings, this does not happen uniformly across days/hours and across users (i.e., some users will create more content than others, and some days of the week will see higher content being produced overall than others). To mimic a realistic behaviour as much as possible, we have extracted content publication rates from users of the content bookmarking website Digg [Digg, 2010]: first, we have selected Digg publication data for period of five months starting August 2007; based on this subset we then filtered users such that every user had published at least 50 messages during this period. We have then ordered Digg users based on the number of publications; we have then extracted 100 users, making sure the original publication rate distribution is preserved.

To conduct our simulations, we needed to combine the mobility, the social network and the publication rate layers together. Let us consider mobility and interest network first: when using the Reality Mining dataset, users in the mobility traces corresponded to actual users in the interest network, so a direct mapping was created. When using the Cabs and the Last.fm datasets, random overlays have been created instead, as no work exists to date describing a realistic way of performing such correlation (the results presented in the next section are averages computed over three random overlays). Let us now consider social network and publication datasets: in this case, we have overlayed the selected Digg users onto the social network users, so that the most popular nodes in the social network were also the most active nodes in the publication dataset; this fits the intuition that users with many followers have more content to share than users with few/no followers. Note that we have also experimented with three random mappings between social network and publication dataset; as the results were consistent with those presented in the following, we do not dwell further in this matter.

5.4.2 Metrics

The goal of CoHabit is to distribute workload fairly amongst nodes, while not compromising delivery. In order to quantify the trade-off between these two aspects, we thus focus on two main metrics of *fairness* and *delivery*. We define fairness as the coefficient of variation of the total load forwarded by nodes, and formulate it as:

$$fairness = 1 - \frac{\sigma}{\mu}, \tag{5.3}$$

where μ is the average number of messages forwarded by nodes over the whole simulation period, σ presents the standard deviation from the mean, and fairness is a value within the range of $(-\infty, 1]$. The coefficient of variation indeed reflects discrimination in the system by showing dispersion amongst population. As the standard deviation σ converges toward zero (i.e., nodes forward the same amount of traffic), the fairness value becomes closer to 1 (i.e., uniform workload distribution); at the opposite extreme, when σ is very high and far from the mean μ (i.e., the load is highly skewed, with only a tiny portion of nodes in charge of the whole content distribution), fairness will drift towards $-\infty$.

For the sake of comparing fairness of our approach with other protocols and allowing future comparisons to our work, we also measure fairness in terms of Jain's Fairness Index [Jain et al., 1984]. Jain's fairness index is a well known fairness metric in the network engineering community, describing the ability of the system to deliver a fair share of load to all the users, and is defined as follows:

$$Jain'sFairness = \frac{(\Sigma x_i)^2}{(n * \Sigma x_i^2)}, \tag{5.4}$$

where n is the number of users and x_i corresponds to the share received by user i (or, in our case, the number of forwarded messages by user i). The Jain's Fairness value lies in the interval $(\frac{1}{n}$, 1].

While previously we investigated delivery of our content dissemination protocol Habit in terms of a ratio of received messages over all the messages wanted by the user, in this work we are not interested in measuring the performance as such. Rather, we want to investigate the *change in delivery* between a realistic setting (where nodes would defeat cooperation if overloaded) and the idealistic setting (where nodes always participate). We thus measure delivery as the *ratio* of messages successfully delivered by CoHabit under different drainage threshold, with respect to Habit in an ideal (albeit unrealistic) consumption-less environment where nodes have infinite battery, formulated as:

$$delivery\ ratio = \frac{\sum m_i}{\sum M_i}, \tag{5.5}$$

where $\sum m_i$ is the sum of messages delivered to the interested users by CoHabit, and $\sum M_i$ is the sum of the messages delivered to interested users by original Habit in a consumption-less environment.

For completeness, we also report the change in delivery for our benchmark protocols, which we define next.

5.4.3 Benchmarks

We compare fairness and delivery performance of CoHabit with two benchmarks: Epidemic [Vahdat and Becker, 2000] and Habit [Mashhadi et al., 2009], which represent upper bounds in terms of fairness and change in delivery respectively.

Epidemic - Although the Epidemic routing protocol has been extensively used as a benchmark in the literature to show comparable delivery of DTN protocols, it can also be thought of as an upper bound to *achievable* fairness. This is because achieving $fairness = 1$ is impossible as the fairness value will inevitably depend, and to a potentially large extent, on the topology of the network. In other words, in any real human mobility setting, those nodes sitting at the edge of the network will never have the opportunity to contribute to the content forwarding protocol as much as central nodes, even if they wished to.

In order to put an upper bound on the *achievable fairness* for a given topology, we study the fairness obtained by a simple epidemic protocol: as this protocol does not *favour* any particular node as carrier (unlike DTN protocols such as [Costa et al., 2008, Yoneki et al., 2007a, Lindgren et al., 2004, Lindgren et al., 2006]), its fairness value is only impacted by the topology of the mobility traces. In the following, we will report both the row fairness values obtained with CoHabit, and fairness ratios with respect to what epidemic can achieve.

Habit - In our evaluations, we use Habit for two different purposes: first, its performance is measured in an consumption-less setting (where participation persists at all time) and used to calculate the *change in delivery* of CoHabit (as we described in Section 5.4.2);

second, its change in delivery is evaluated under a realistic setting (where nodes have limited battery) and is assigned to act as a lower bound benchmark to CoHabit, assisting us with demonstrating CoHabit's performance gain due to its careful distribution of load.

Furthermore, we compare CoHabit's performance with the only major protocol which aims to achieve fairness in DTNs, FairRoute [Pujol et al., 2009], which was earlier described in Section 5.1.2. In so doing, we had to adapt FairRoute to our simulation environment as described next.

FairRoute - FairRoute protocol was originally evaluated under a one-to-one communication setting where each node in the network issues a message for every other node, over a period of six months, thus considering a scenario where each message has only one intended recipient; furthermore publication was a rare event. However, in order to claim a protocol to be fair in distributing workload, it is essential to analyse its performance under a more heavily loaded network configuration (i.e., nodes frequently publishing messages, each of which may have many different recipients).

Therefore, to evaluate FairRoute in such a loaded simulation environment, we have taken the following steps: first, we have evaluated it in scenarios where the communication is one-to-many, allowing messages to have multiple destinations based on the publisher's social network. Second, the publication rate is changed to incorporate properties observed from Web 2.0 applications, thus corresponding to a more realistic rate of publications. To cater for the former, we have assumed that nodes are aware of their full interest network, allowing them to identify all the nodes who are interested in receiving content from them. Equivalently, in the case of CoHabit, this is done by nodes propagating their interest network during the training period. To cater for the latter step, we have evaluated FairRoute using Digg publication dataset as was previously described in Section 5.4.1.

Finally, in order to have an impartial comparison between CoHabit and FairRoute, we apply the same resource constraints to the nodes in the network. To do so, we require introducing the time dimension to FairRoute, as its original implementation neglects time and defines fairness only in terms of the overall number of forwards over the simulation time as a whole. Therefore, we added a component to the original protocol which locally monitors, at each node the number of forwarded messages in Δt time, as it is done in CoHabit. In this manner, nodes can stop participation after forwarding a certain amount of messages during Δt time (i.e., drainage threshold).

5.4.4 Parameters

Table 5.2 reports our parameter settings for the performed evaluations. We omit reporting further parameters required by the underlying Habit protocol, as they are same as those previously reported in Section 4.4.4. Note that a different time-to-live is used in the two datasets, in order to cater for their topological properties, thus reflecting the shorter inter-contact time for Cabs, in comparison to the longer inter-contact time of Reality Mining traces. We have conducted a much broader set of experiments by varying time-to-live of messages; however we only report representative results based on experiments conducted using the following parameters.

	Reality Mining	Cabs
Mapped Social Network	Inferred From Phone Calls	Last.fm
Publication Dataset	Digg	Digg
Number of nodes	96	100
Training Period	35 days	21 days
Simulation Period	150 days	42 days
Time-to-live of the messages	10 days	3 days
Critical Boundary	70%	70%

Table 5.2: CoHabit Simulation Parameters

In the next section, we use ‘Cabs’ to concisely refer the dataset comprising cabs movement, Last.fm social network, and Digg publication rate (where messages have time-to-live= 3 days). Similarly, we use ‘Reality Mining’ to refer to the dataset comprising Reality Mining movement and social network, in combination with the Digg publication, and in a setting whereby messages are valid for 10 days.

5.5 Results

5.5.1 Sensitivity Analysis

In this section we analyse the effect of protocol-specific variables, such as drainage threshold. We used these results to tune these variables for the rest of the evaluations. In so doing, we turn our attention to the Cabs scenario, which allows us to stress-test CoHabit under heavy load. Similar results have been obtained for the Reality Mining setup, although in a smaller scale. In order to present CoHabit's fairness in ratio to the proposed upper bound benchmark (Section 5.4.3), we first execute the Epidemic protocol on the Cabs traces, with the parameters summarised in Table 5.2 and nodes having no power constraints (i.e., nodes always forward any messages given to them). By performing this experiment, we were able to measure a fairness value of 0.61, which presents the highest achievable fairness, for this setting. Thus, in the following experiments we will report both row fairness values obtained with CoHabit, and fairness ratios with respect to this upper bound.

Effect of the Load-Aware Component. In Section 5.3, we defined α as a weight to tune the importance given to the load-aware component of the protocol. In this section, as a first experiment, we measure the impact of different values of α on both delivery and fairness, when fixing the drainage threshold to 50 messages over 5 days. The results are reported in Table 5.3. A value of $\alpha = 0$ means that routes are constructed at the source purely based on cost (interest-awareness component); the fairness achieved thus only depends on the *load alleviation* component. When $\alpha = 1$ instead, routes are source-computed based on their estimated load only. As expected, when $\alpha = 0$ fairness is significantly reduced; moreover, delivery is compromised, as messages continue to queue up (and eventually expire) on heavily congested nodes, on which the load alleviation component is continuously called upon. As more weight is given to load reasoning (i.e., increasing α), the load is more evenly distributed, leading to considerably higher fairness. Note that with $\alpha = 1$ delivery is slightly worse than with $\alpha = 0.75$. This is because paths are more conservatively constructed at source, choosing longer routes which result in some messages to expire; this is particularly the case for the very first few messages published, when nodes have not yet learned enough about the actual load distribution in the network, thus making unnecessary cautious choices. We will revisit the delay dimension of the load-aware reasoning in the next section.

	Delivery	Fairness (ratio to upper bound)
$\alpha = 1$ (Load only)	81.47%	0.51 (0.83)
$\alpha = 0.75$	81.63%	0.46 (0.75)
$\alpha = 0$ (Cost only)	77.90%	0.33 (0.54)

Table 5.3: CoHabit Fairness for Different α Values (Drainage Threshold = $\frac{50msgs}{5days}$)

Effect of the Drainage Threshold. As a second experiment, we have evaluated the effect of the drainage threshold on fairness. Table 5.4 presents the results for $\alpha = 1$ (i.e., the setting where maximum weight is given to load, and thus the highest fairness is achieved). A high threshold is representative of applications that share small messages (e.g., Twitter posts); in this case, more messages can be exchanged before battery drains, and nodes can be more relaxed about the workload distribution. When applications share bulkier messages instead (e.g., images and videos), each message forwarded has a high impact on the node's power consumption (small threshold); in such scenario, nodes will not tolerate uneven workload, and fairness becomes a primary objective. As shown, CoHabit offers a fairness as high as 95% of that offered by epidemic in these circumstances.

Drainage Threshold ($\alpha = 1$)	Fairness (ratio to upper bound)
$\frac{25msgs}{5days}$ (e.g., Video files)	0.58 (0.95)
$\frac{50msgs}{5days}$ (e.g., Music files)	0.51 (0.83)
$\frac{100msgs}{5days}$ (e.g., Text files)	0.41 (0.67)

Table 5.4: CoHabit Fairness for Different Drainage Thresholds

Based on the above analysis, we grant the α value to be 0.75 for the rest of evaluations, as it offers a high fairness value without over cautious routing which can result in losing in delivery. We also set the drainage threshold to 50 messages over 5 days for the remaining experiments, when using the Cabs setting.

5.5.2 Benchmark Analysis

In this section, we report the results of the conducted benchmark analysis for CoHabit's performance against Habit. We draw our attention back to the problem demonstrated in Section 5.1.1 where we showed Habit (and CDC) causes the participatory network to collapse due to an unfair load distribution (Figure 5.1), which in turn causes some nodes to quickly exhaust their battery reserves (Figure 5.2). We now illustrate how CoHabit improves nodes participation in the same context (Cabs traces with drainage threshold of 50 messages over 5 days). In particular, Figures 5.5 and 5.6 illustrate CoHabit's fairness (with $\alpha = 0.75$ as concluded from sensitivity analysis) by showing load distribution amongst nodes, as well as nodes' life expectancy, respectively. Moreover, they present the results for Habit (with and without unlimited battery respectively), under the Cabs setting. Let us first focus on load distribution presented in Figure 5.5. Note that, in the case of Habit, in order to observe the distribution of load amongst users, we have given nodes infinite battery (idealistic case), otherwise they would all reach the drainage threshold causing the full content distribution network to collapse. While with Habit (with unlimited battery setup) this distribution is heavy-tailed (with a few nodes doing all the work, and many nodes contributing almost nothing), in CoHabit (and realistic battery availability), the load is rather evenly spread. However, this more uniform distribution of load comes with trading off on delivery. Indeed CoHabit is only capable of delivering 81% of the messages that Habit with *un*limited battery can achieve (based on Table 5.3), thus, trading off the other 20% delivery to achieve a fair load distribution.

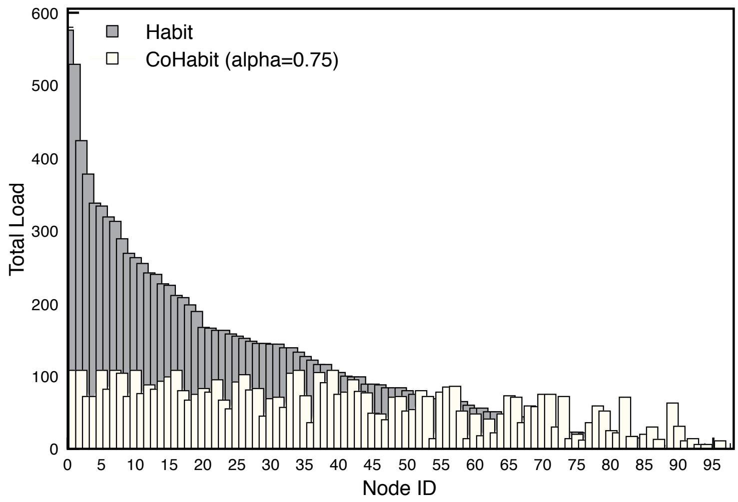


Figure 5.5: CoHabit and Habit's Load Distribution

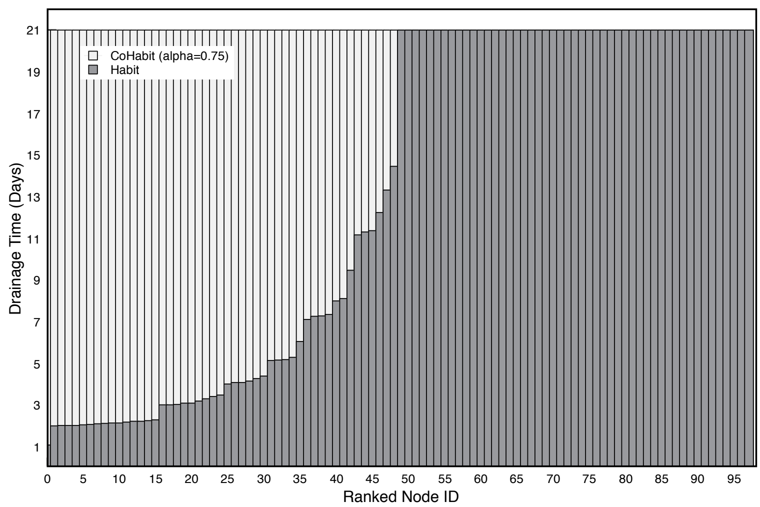


Figure 5.6: CoHabit vs. Habit Drainage Time

We have then enforced a realistic setting by removing the assumption of *un*limited battery for Habit, and repeated the experiment with a drainage threshold for both *Habit* and CoHabit of 50 messages over 5 days (as done in Figure 5.2). As Figure 5.6 illustrates, half of the nodes running Habit stop participation at some point in the simulation (and one third of them in the very first couple of days); this causes the delivery of only 63% of these messages in the consumption-less setting. On the contrary, all nodes running CoHabit continue contributing to the delivery for the whole duration of the simulation, demonstrating the effectiveness of the

Load Prevention and Alleviation components in preventing unfair workload distribution and thus maintaining participation in the network.

So far we have concentrated on CoHabit's performance in terms of the trade-off between delivery and fairness; we now turn our attention towards *delay*, another metric which can be highly impacted by load-aware path reasoning. We measure the delay that CoHabit introduces to message delivery by avoiding central nodes, in terms of both time and number of hops in the selected paths. Figure 5.7 illustrates the hop distribution for CoHabit in relation to Habit (with *un*limited battery), for both Reality Mining and Cabs settings, where the x-$axis$ presents the longer (positive values) or shorter (the negative values) paths selected by CoHabit. The result shows that, in the case of Reality Mining, CoHabit delivers $\approx 70\%$ of messages by taking longer (or same length) paths. This is expected as the Reality Mining traces are less connected and reflect scenarios with sparser mobility traces, thus CoHabit avoids the loaded/central nodes by detouring through paths consisting of more hops. However, in the case of Cabs, CoHabit delivers $\approx 80\%$ of messages via shorter (or same length) paths than Habit. This is because of the topological properties of the Cabs traces (i.e, frequent encounters), allowing CoHabit to find alternative paths of same/shorter length, while still avoiding the central nodes.

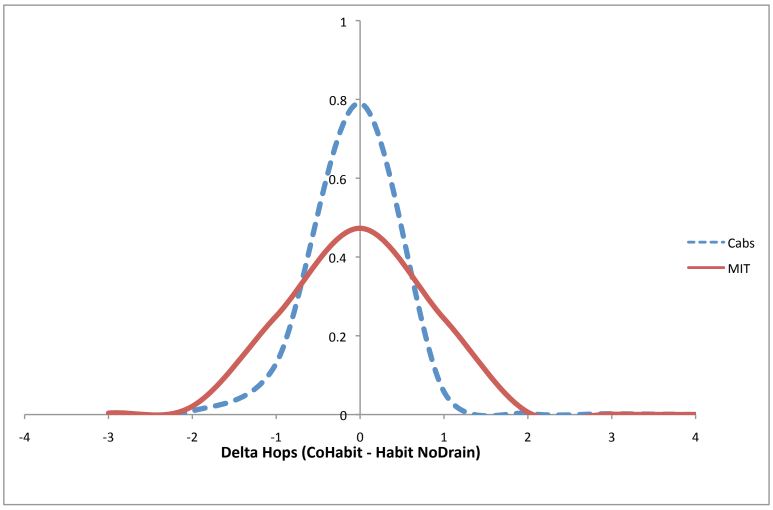
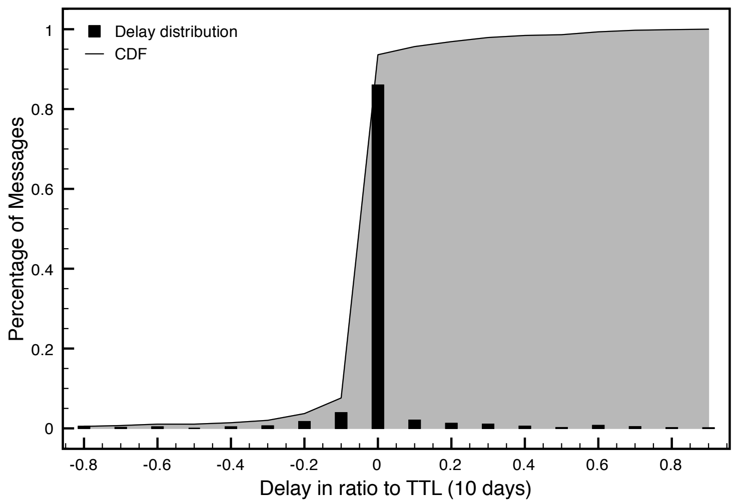
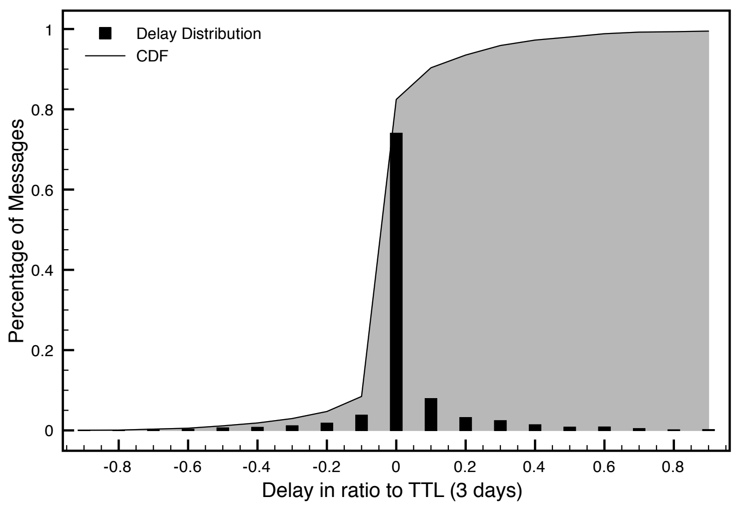


Figure 5.7: Hop Distribution for CoHabit on Reality Mining and Cabs Traces

However, travelling the same number of hops does not necessarily imply that the delivery is achieved within the time duration. Therefore we have measured the delay of CoHabit in comparison to Habit, by computing a delta delivery time for messages that were delivered by both protocols. Figure 5.8 presents the delay distribution for all the commonly delivered messages for Reality Mining and Cabs respectively. It is worth noting that we omit counting expired messages in the presented results, as their effect is already quantified in the delivery



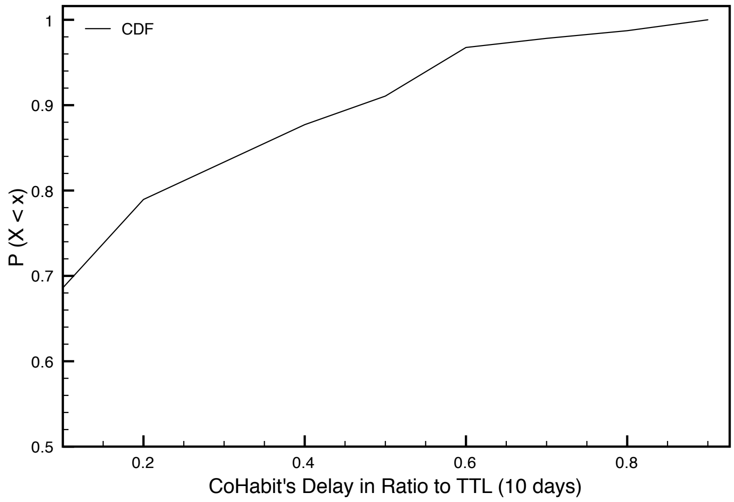
(a) Reality Mining



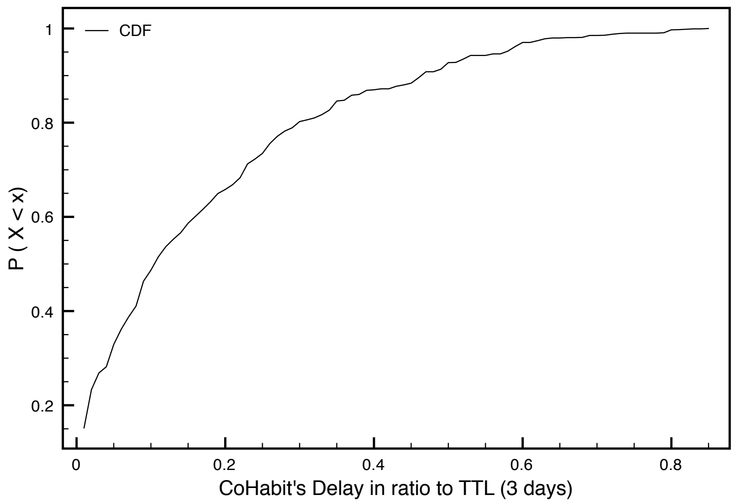
(b) Cabs

Figure 5.8: Overall Delay Distribution for CoHabit's Performance

metric (i.e., causing a reduction in delivery). Therefore, the figure illustrates time analysis for the *successfully delivered* messages only. Those that were delayed (with respect to Habit) are shown in the positive side of the x-$axis$, while those that were delivered faster are shown on the negative side of the x-$axis$. Furthermore, the delay was measured in relation to the time-to-live of messages, which was set to 3 days for Cabs and 10 days for the Reality Mining setting. As it is observed in both settings (Figure 5.8a and Figure 5.8b), whilst there are messages that experienced delay, CoHabit delivers most of the messages (80% in case of Reality Mining, and 70% in case of Cabs) within the same time duration as Habit. Also, there is a small percentage of the messages in both Reality Mining and Cabs setting that were delivered faster by CoHabit, due to different path choices resulting in faster or shorter paths, similar to the observation in Figure 5.7.



(a) Reality Mining



(b) Cabs

Figure 5.9: Delay Distribution for CoHabit

We have further studied the delay distribution of those 20% delayed messages. In this regard, Figure 5.9 expands on Figure 5.8 by presenting the distribution for the delayed messages only (zooming into the positive side of Figure 5.8). Let us first consider the Reality Mining case: Figure 5.9a illustrates that 80% of the delayed messages were only delayed within 2 days (0.2 of 10 days time-to-live). Similarly, in the case of Cabs (Figure 5.9b), 80% of the delayed messages were indeed delayed for less than a day. Furthermore, the growth rate of distribution implies that messages were relatively longer delayed in Cabs traces than those in Reality Mining. This is because in the case of the Cabs setting, there exists a denser interest network causing each message to be intended for more number of recipients in the network. With reference to Figure 5.7, we can conclude that in the case of a heavier loaded network (i.e., such as Cabs where the mobility traces are well connected and the interest network is denser), while the messages often traverse a shorter path, the heavy load in the network causes them to be delayed for longer unlike in sparse networks such as Reality Mining. However, the increase in delivery time is an acceptable trade-off for achieving a fair distribution of messages, as the content is not time critical yet still delivered within the accepted time-to-live.

5.5.3 Comparative Evaluation

Finally, we report the results obtained when comparing CoHabit with FairRoute [Pujol et al., 2009], the only other major DTN load-balancing protocol. In so doing, we first analyse FairRoute's performance in our described setting. We do this by adapting FairRoute to our simulation environment as we described earlier in Section 5.4.3. Similar to the originally reported evaluation, we also evaluate FairRoute on Reality Mining mobility traces, thus preserving the trace-driven parameters presented in [Pujol et al., 2009], but using the social and publication datasets described in Section 5.4.3. Moreover, in these experiments, we compare FairRoute against CoHabit under the same participatory condition, that is by setting the very same drainage threshold to nodes. Figure 5.10 illustrates the life expectancy of nodes using FairRoute when the drainage threshold is set to 50 messages per 5 days. As depicted, almost half of the network ceases participation during the first day, reflecting the aggressive forwarding behaviour of FairRoute.

We have then turned our attention to FairRoute's ability of delivering messages by computing the *change in delivery* for FairRoute (i.e., recall this metric is computed as the ratio of messages delivered in an energy-constrained setting, with respect to those delivered by the *same protocol* with no constraints). In so doing, we observed a very poor performance in terms of delivery as only 2% of the messages that were delivered in an unlimited-energy setting managed to reach their destinations when the resource restriction was applied to FairRoute. This is not surprising, as almost half of the network reached the drainage threshold and thus stopped participation during the first couple of days (Figure 5.10). In order to make FairRoute's comparable to our work, we integrated our proposed *Load Alleviation* to the original FairRoute. In this manner, we enable FairRoute to prevent nodes from ceasing participation altogether, by allowing them to take a temporary step back from exhausting their resources. Figure 5.11

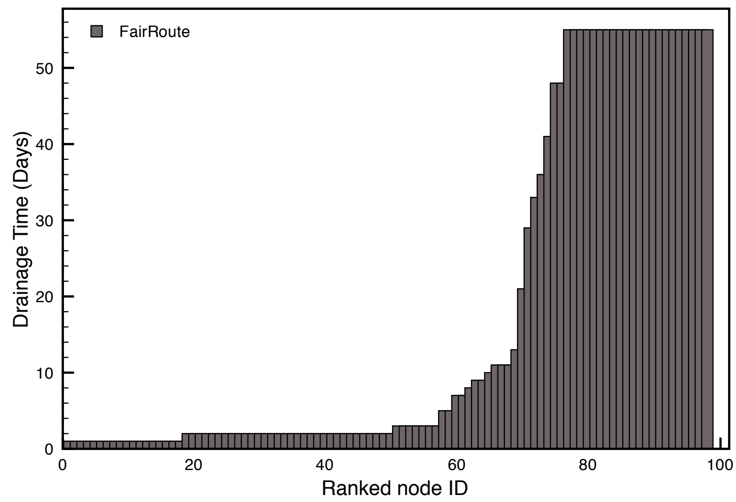


Figure 5.10: FairRoute's Drainage Time

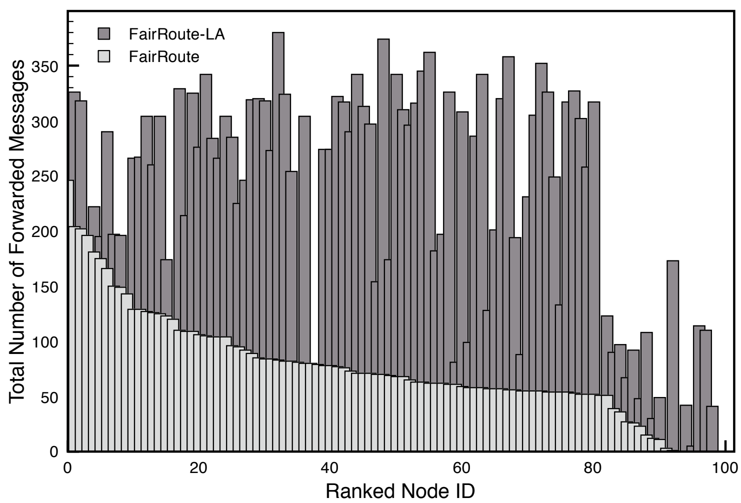


Figure 5.11: FairRoute Load Distribution (With and Without Load Alleviation Component)

presents the load distribution for FairRoute, with and without load alleviation component. As it is observed, when merged with load alleviation component, FairRoute manages to forward more messages as nodes only stop participation temporarily when critically loaded.

To assess the effect of this integration, we have then measured the Jain's Fairness Index for both FairRoute with (FairRoute-LA), and without load alleviation. Table 5.5 reports on this results for various participatory conditions (i.e., drainage thresholds). As expected, once integrated with the load alleviation, FairRoute achieves a higher fairness than original FairRoute, for all the values of the drainage threshold. This is because nodes step back from forwarding once they reach a certain amount of forwarded messages, allowing them to adjust their contribution to the delivery network.

Drainage Threshold	$\frac{50msgs}{5days}$	$\frac{100msgs}{5days}$	$\frac{150msgs}{5days}$
FairRoute without Load Alleviation	72.5%	69.8%	69%
FairRoute with Load Alleviation	81%	79.9 %	78.8 %

Table 5.5: Jain's Fairness Index for FairRoute With and Without Load Alleviation Component

Based on the reported analysis of FairRoute, we only consider FairRoute integrated with load alleviation component for the rest of our comparative evaluations, and refer to it as *FairRoute-LA*. This is to allow us to draw a sound conclusion in an equal network condition where nodes do not cease participation. For completeness, we also report on the performance of Habit and CDC protocols. Table 5.6 presents the change in delivery for CoHabit and other benchmark protocols in a constrained participatory network when using Reality Mining traces. As the drainage threshold increases, the delivery increases due to nodes forwarding more messages before exhausting their battery/entering the load alleviation. However, there are remarkable differences to be observed: FairRoute-LA (with load alleviation) is the protocol that suffers the highest loss in message delivery, with at most 27% of messages delivered (with respect to the infinite-battery scenario). This low performance is caused by the aggressive forwarding strategy of FairRoute, which causes nodes to quickly reach the critical boundary and step back from participation until sufficient time has passed (e.g., they have recharged their battery). In addition, FairRoute has been designed based on the idea of offloading messages from central nodes to least loaded intermediaries, causing the network delivery to entirely depend on the forwards made by the central nodes, and thus collapsing once they temporary stop forwarding. Similar results were obtained for experiments conducted using Cabs setting.

Drainage Threshold	$\frac{50msgs}{5days}$	$\frac{100msgs}{5days}$	$\frac{150msgs}{5days}$
CDC	41.83 %	43.77 %	46.74%
Habit	22.66%	39.18%	46.14 %
FairRoute (With Load Alleviation)	9.13%	18.80%	27.9%
CoHabit	57.88%	66.57%	72.73%

Table 5.6: Delivery Achieved for Reality Mining Traces

While less severe than FairRoute-LA, CDC and Habit suffer a neat reduction in delivery too. CoHabit exhibits the highest delivery across all benchmark protocols instead: in the very same situation where CDC and Habit achieve only 46% delivery (150 message in 5 days threshold), CoHabit achieves 72%. The CoHabit load-aware mechanism is thus capable of redirecting traffic toward less-used part of the network, offering nodes the ability to recover when becoming overloaded. There is still a portion of messages that even CoHabit is not able to bring to destination within their time-to-live. This is because some messages are set to follow longer paths (as we showed earlier) caused by the load prevention phase.

To investigate these results in more details, Figure 5.12 depicts the cumulative number of messages delivered in the system over time, for a threshold of 50 messages over 5 days, using the Reality Mining traces. As can be seen, FairRoute-LA's, Habit's and CDC's delivery slows down significantly very early into the simulation; this is because the most central nodes in the network have quickly reached their drainage threshold and ceased participation, either temporarily (in case of FairRoute-LA) or altogether. After 60 days, the curve goes completely flat, and no further deliveries are accomplished. In comparison, CoHabit manages to keep up delivering messages throughout, effectively finding fairer paths to destinations.

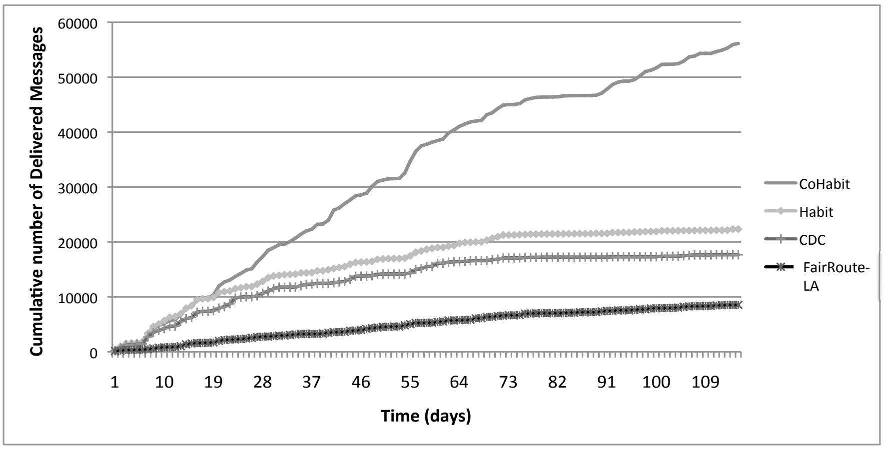


Figure 5.12: Cumulative Delivery for Reality Mining

We also compare CoHabit and FairRoute-LA in terms of achieved Jain's fairness. We omit these values for CDC and Habit, as they were not designed to maintain a fair distribution of load. Table 5.7 reports the measured Jain's Fairness Index in the Reality Mining setting. While FairRoute-LA appears to achieve a considerably high fairness, this uniform distribution of workload comes at the expense of delivery (Table 5.6). More precisely, when CoHabit fairness is almost 20% less than FairRoute-LA, its delivery is significantly higher (50% more) for the same drainage threshold (50 messages in 5 days). Indeed, if we combine these results with those reporting change in delivery in Table 5.6, we can conclude that CoHabit is capable of distributing the load more equally amongst the participants, and unlike FairRoute-LA (and original FairRoute), without compromising on delivery.

Drainage Threshold	$\frac{50msgs}{5days}$	$\frac{100msgs}{5days}$	$\frac{150msgs}{5days}$
FairRoute (With Load Alleviation)	81.00%	79.90%	78.80%
CoHabit	65.00%	60.31%	56.35%

Table 5.7: Jain's Fairness Index for CoHabit and FairRoute

5.6 Conclusion

In this chapter, we tackled the problem of participation in content dissemination networks, by modelling users participation in terms of their resource contribution to forwarding messages in the network. We particularly focused on battery constraints of mobile devices, which impose a limit on the number of messages a user can forward in a content delivery network. We proposed a load-balancing model which, once integrated with a source-based routing protocol, achieves a more uniform workload distribution amongst participants, without compromising delivery. In so doing, each node locally estimates the load of other nodes in the network and uses this information to select the paths that are less loaded (i.e., load prevention). Moreover, each node locally monitors how much traffic it has been forwarding, so that, should a critical limit be reached, it can temporary cease participation (i.e., load alleviation).

We implemented the proposed approach on top of the Habit source-based DTN protocol, and evaluated this realisation by means of network simulations. To draw a sound conclusion on the performance of our load-aware protocol (CoHabit), we evaluated it under realistic network settings where users publication and social network corresponds to the real behaviour observed from use of Web 2.0 applications.

We compared the performance of CoHabit with Epidemic as a benchmark for fairness, and Habit as an upper bound benchmark for delivery. We drew a comparative evaluation of CoHabit with state-of-the-art DTN protocols. In particular, we compared CoHabit against FairRoute [Pujol et al., 2009], and showed that both prevention and alleviation components are necessary to achieve high delivery and fairness. We did so by integrating FairRoute with our load alleviation component and showed that, when the underlying protocol fails to prevent the main nodes from becoming overloaded, alleviation by itself is not enough. Indeed our results presented that FairRoute-LA suffers from extensive loss in delivery.

In the next chapter, we turn our attention to the produced content. So far we considered all content produced in the network to be equally wanted by end-users; however, in practice this is not the case, as users value each piece of content differently. The question that arises here is thus, *what messages to forward* given that limited battery will not allow the delivery of them all. We investigate this problem next.

Chapter 6

Priority Scheduling

In participatory DTNs, users' satisfaction is not only based on the number of relevant messages they receive, but more importantly it is based on the value they attach to the received messages.

In this chapter, we investigate the problem of prioritising content in participatory networks where nodes contribution to content delivery network is limited (Ch 4 as described in Chapter 1). In particular, we lift the assumption that all messages are worth the same, and propose a priority scheduling framework which allocates the scarce resources available on devices to deliver those most wanted (i.e., most valued by the end-users). In so doing, our approach leverages knowledge from both the *physical* and the *application* layer, whereby messages are scheduled to be forwarded based on a combination of the likelihood of future encounters (physical layer) and the value that end-users attach to such messages (application layer).

The rest of this chapter is structured as follows: we first provide a background to the forwarding mechanisms of the state-of-the-art protocols, describing the challenges faced by prioritisation (Section 6.1). We then propose a conceptual model for prioritising messages in a participatory content dissemination network (Section 6.2), before realising it on top of a DTN routing protocol (Section 6.3). We evaluate our priority scheduling framework by means of simulation using a variety of real traces, whose settings are described in Section 6.4. In Section 6.5, we report on the performance results of our prioritisation scheme. Finally, we summarise our findings and conclude this chapter in Section 6.6.

6.1 Background

The latest generation of DTN protocols [Pujol et al., 2009, Balasubramanian et al., 2007, Mashhadi et al., 2011] acknowledge the fact that a limit must be placed on the amount of resources (e.g., battery or storage) that nodes are willing to share, consequently reducing the number of messages that intermediaries can forward at any point of time.

Common to all approaches proposed so far is the treatment of messages as if they were all worth the same to end-users: the decision of what message to forward next, in the hop-by-hop path from source to destination, is entirely driven by the next physical encounter, in a sort of *first-encountered/first-forwarded* basis. In other words, if an intermediary node has enough allocated resources to forward one message only, it will forward the message whose next hop node (be it destination or another intermediary in the path towards the destination) is met first. However, end-uers are not equally interested in all messages, and an encounter-based forwarding approach can indeed allocate scarce resources of intermediary nodes for forwarding messages that are less desired by end-users network-wide. To better appreciate the challenge we tackle, let us consider the following scenario where an intermediary node u_i is responsible for forwarding two pieces of content c_1 and c_2 wrapped in messages m_1 and m_2 respectively. Let us assume in this scenario that c_1 is highly desired by its end-users (i.e., the identified destinations for message m_1 based on information provided by application layer), while c_2 is far less desired. Let us also assume that u_i's resources allocated to the content dissemination application are running out, so that it can only forward one more message in the current time period Δt (e.g., within the next day). The question that arises here is *which message should node u_i forward given the insufficiency of its resources?*

Two alternative approaches could be followed: on one hand, we could let the *physical network* drive the forwarding step entirely. For example, if the next hop node for m_2 is encountered first, then the message m_2 would be forwarded, at the expense of end-users awaiting for m_1 (and its higher-value). In other words, first-encountered/first-forwarded protocols may cause messages of little value to use up the scarce resources available, at the expense of highly-valued messages; note that this is the approach used by state-of-the-art DTN protocols [Mashhadi et al., 2009, Musolesi and Mascolo, 2008, Costa et al., 2008, Pujol et al., 2009, Daly and Haahr, 2007].

On the other hand, we could let the *application layer* drive the forwarding decision by adapting a *highest-value/first-forwarded* approach. In this case, node u_i would reserve its remaining forward allowance for m_1; however, next hop node for m_1 may not be encountered for another couple of days, during which u_i's resources could be reset (for instance by re-charging the device) and thus its forwarding allowance increased. Not forwarding m_2 when the opportunity raises may thus result in unnecessarily missed deliveries.

Nodes participating in a DTN must thus be able to allocate the scarce resources available for forwarding messages of high value, whilst also not compromising delivery due to missed opportunities. To do so, we present next a priority scheduling approach, which reasons upon nodes' mobility patterns (physical layer) *and* messages' values (application layer) to achieve high end-user satisfaction without cutting back on delivery.

6.2 Conceptual Model

Our approach to priority scheduling relies on information from the application and the physical layer. More precisely: first, information from the application layer is exploited in order to quantify how much a piece of content is valued within the network; second, the physical layer is consulted so to provide an estimate of encountering likelihood of next hop node of messages in the queue. These two sets of information are then combined to create a novel prioritisation scheduling framework which, once integrated with an underlying DTN routing protocol, achieves high satisfaction without compromising delivery.

6.2.1 Modelling the Content

In order to design a content distribution protocol that prioritises messages' delivery based on their value to end-users, we first need to quantify what this 'value' is. That is, how much each message is wanted across the network. Grounding on our model formulation defined in Chapter 3, a dual way of modelling user interest can be used so as to define the weight of each interest relation. This can be done by modelling a *weighted* user social network, in the people-centric approaches, whereby Alice's profile not only states she wants to receive messages from Bob, but also how much she wants the content c published by Bob, $w_c \in (0, 1]$. These weights can either be explicitly defined by users, as it is done in some Web 2.0 applications (such as Rummble[1]), or implicitly derived by looking, for example, at the frequency of interaction between users (e.g., in Twitter, the frequency of `@username` directed messages).

Similarly, in information-centric approaches, where we have users stating their topics of interests using tags, we require weights associated to each so to describe how much a tag t_i is valued by the user. These weights are defined either explicitly through ranking by the user, or implicitly by monitoring user's tag usage (e.g., which tags were used most to described their published or bookmarked content). In this case, a piece of content c is wanted by the end-user u_i with weight w_c, whereby w_c represents the weighted average of tags in T_c (i.e., describing c). Figure 6.1 depicts examples for both interest models, with u_i being equally (and maximally) interested in users u_r, u_x, u_y and u_z ($w = 1$) in the case of the people-centric model, and interested in tags t_1, t_2, t_3 and t_4 when interest is modelled in an information-centric way; user u_j has expressed interest in two sources u_p and u_q, but is more interested in content produced by u_p ($w = 0.8$) than in u_q ($w = 0.2$).

Note that, in any prioritisation scheme, users who have many social connections and who

[1] www.rummble.com

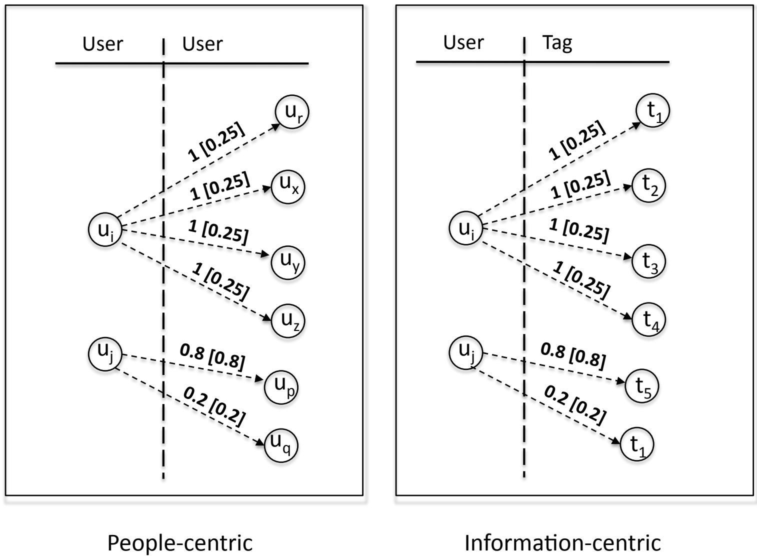


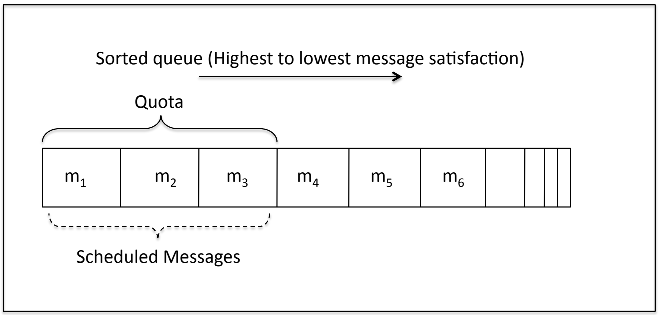
Figure 6.1: Weighted Interest Profiles of Users u_i and u_j

are equally and maximally interested in all of them (e.g., user u_i in the above example) risk driving the whole content distribution network to work for them, at the expense of other nodes (e.g., user u_j) who may have less social connections and/or of different values. Therefore, in our model, the value of a message is not simply the (explicit or implicit) weight in the interest profile, but such weight divided by the sum $\sum w$ of all weighted edges departing from u_i in the social graph. With reference to Figure 6.1, messages produced by u_r, u_x, u_y and u_z would thus have a value for u_i of $w/\sum w = 1/(1+1+1+1) = 0.25$, while messages produced by u_p would have a value for u_j of $0.8/(0.8+0.2) = 0.8$. This processing aims to promote a fair share of network resources to be used in support of every single participant; protecting against malicious and adversarial behaviours is outside the scope of this thesis (as stated in Chapter 1).

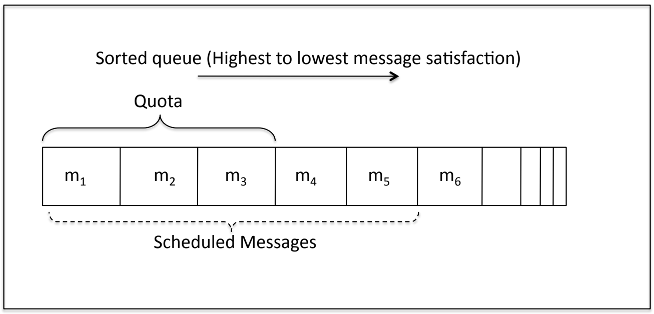
Furthermore, we assume that the value of messages remains the same throughout their validity period (i.e., before they expire). In other words, as messages in the network age, their importance value to end-users does not decay. The reason for this assumption is twofold: first, in our proposed scenario (in Chapter 3), the content that is shared in the network is of *leisure* type and delay tolerant. Second, this thesis main objective is to bridge the gap in DTNs research by accounting for user participation, hence throughout this thesis reducing delay has not been our primary objective. Thus, we do not further dwell into this matter and assume that users value a piece of content uniformly throughout its lifetime.

6.2.2 Priority Scheduling

In order to prioritise messages without compromising the delivery due to missed opportunities, we require a scheduling framework which incorporates information about node encounters from



(a) Fully Connected Network



(b) Human Network

Figure 6.2: Dynamic Bundle

the physical layer. To appreciate the challenges faced by the opportunistic nature of delay tolerant networks, let us consider a scenario where nodes are connected at all times. Let us further assume that, in this scenario, a node in any given time period Δt can forward at maximum 3 messages. We call this value $quota$, where $quota$ represents the maximum number of messages that the node can forward within a given time period Δt, and it depends on, for example, the portion of local resources that the node is willing to allocate to the content delivery network. In this case, because of stable network connection, the probability of delivering any message to its intended recipient is always 100%. In such cases, it thus suffices to rely on information from the application layer to schedule forwarding of messages. In other words, the application layer provides enough information (i.e., message values) to *sort* messages in order of priority, and messages can thus be scheduled to be sent in the very same order they appear in the sorted queue (i.e., the top $quota$). Figure 6.2a illustrates this example.

Let us now consider the case of a human DTN, where connection between nodes is opportunistic, yet predictable [Rhee et al., 2008, Song et al., 2010]. In this case, scheduling only the top three messages could be a waste: with reference to Figure 6.2b, the node would not attempt to deliver m_4 before m_1, even if an encounter occurred that would enable that. While this guarantees that there will be enough spare resources to deliver higher-valued message m_1

when the relevant encounter occurs, it may also result in an unnecessary missed opportunity, for example, if the node's next encounter with the recipient of m_1 is unlikely to occur within this Δt.

To address this challenge, we propose a priority scheduling framework in DTNs which exploits information from the application layer (i.e., message values) to *sort* messages in order of priority, thus guaranteeing faster processing for high priority ones; information from the physical layer is exploited (i.e., probability of physical encounters) to dynamically *adapt* the number of messages that are currently scheduled for forwarding, in an attempt to minimise the risk of wasting resources because of missed opportunities. In so doing, each node participating in the content distribution network stores messages yet to be delivered in a queue, which is kept *sorted* by decreasing message value. For each message m_i in the queue, $Prob_{m_i}$, which is an estimate of the likelihood of encountering m_i's next-hop recipient within next Δt is computed. The next message to be forwarded can be *any* belonging to the head of the queue; such head does not simply refer to the first $quota$ messages. Rather, it refers to all messages within a *dynamic bundle*, whose size varies depending on the probability of encountering the recipients (or next-hop carriers) of the stored messages. We assume this prioritisation scheme to be deployed on top of an existing human-based DTN routing protocol (e.g., [Mashhadi et al., 2009, Mashhadi et al., 2011, Costa et al., 2008]), whereby past node encounters are logged and processed to compute these probabilities. In designing this approach, we draw inspiration from the TCP flow control mechanism: while TCP uses a *sliding window* to adjust the transmission rate of packets, based on the observed drop rate, we use a *dynamic bundle* to adjust the scheduling of messages, based on the observed encounter predictability. In our approach, we thus define the behaviour of the dynamic bundle based on the encountering probability of each message $Prob_{m_i}$ as follow:

$$bundle.size = \begin{cases} queue.size(), \text{ if } queue.size() \leq quota \\ n \text{ otherwise} \end{cases} \tag{6.1}$$

with

$$n = \left\{ \#m_i | \sum_{i=1}^{queue.size()} Prob_{m_i} \leq quota \right\}$$

In other words, if the queue currently stores fewer messages than the quota allows, all of them can be scheduled for forwarding. Otherwise, the bundle size is set to be equal to the maximum number of messages (n) for which the sum of the probability of them being forwarded within Δt does not exceed the $quota$. Let us look back at Figure 6.2b, and assume the encounter probabilities for the messages in the queue to be: $Prob_{m_1} = 0.2$, $Prob_{m_2} = 0.6$, $Prob_{m_3} = 0.5$, $Prob_{m_4} = 0.8$, $Prob_{m_5} = 0.7$, $Prob_{m_6} = 0.9$. In this case, setting $bundle.size = quota = 3$ is likely to result in missed opportunities, especially because the message at the top of the queue has very little chance of being delivered within this Δt, at the expense of messages m_4 and m_5, whose value is lower (they are further down in the queue) but whose delivery probability is very high. Our approach would thus set the bundle size to include the top $n = 5$ messages, whose aggregated delivery probability does not exceed $quota$ (i.e., $0.2 + 0.6 + 0.5 + 0.8 + 0.7 \leq 3$). This does not mean that more messages will be forwarded than what $quota$ allows; rather, it means that the $quota$ messages to be forwarded in the current Δt can be *any* of the n at the head of the queue, with $n \geq quota$.

No matter what prediction technique is used by the underlying DTN routing protocol to estimate encounter probability, human mobility carries an inevitable degree of uncertainty with it [Song et al., 2010]. In case of high prediction error, the sizing of the dynamic bundle defined by formula 6.1 could be either too cautious (when actual encounters happen less frequently than predicted, causing missed opportunities), or too aggressive (when actual encounters happen sooner than expected, causing more important messages not to be forwarded because of resources being drained on less important ones). We present how we cater for this uncertainty, in the next section.

6.3 Realisation

In order to evaluate the effectiveness of our proposed priority scheduling framework, we require an underlying routing protocol which can estimate the likelihood of future encounters. In this regard, we have selected CoHabit, the fair routing protocol we previously described in Chapter 5. CoHabit relies on regularity prediction in order to select the most probable paths in terms of delivery, these predictions can thus be exploited by the prioritisation scheduling framework so to estimate the likelihood of future encounters of next-hop carriers for the messages in the queue. CoHabit also has a load monitoring component, so by embedding this prioritisation scheduling framework on top of it, we can prioritise messages while still accounting for a fair usage of resources.

To realise our prioritisation scheduling framework on CoHabit, we need to redefine the weight corresponding to the value that each piece of content offers, so to correspond to the satisfaction a *message* offers to all its destinations within the path it is travelling. In fact, in Habit/CoHabit, a message travelling from user u_a to u_b may traverse multiple intended recipients (as our quest to minimise uninterested relayers). In other words, the path that a single message follows typically has multiple recipients in it, so it is possible for the source to compute the value that the published message offers to its recipients along the path, by reasoning on its local view of the interest network. This value is then recorded in the header of the message as part of its metadata, enabling intermediaries to use this value in priority scheduling. It is worth noting that, based on this realisation, two different messages m_1 and m_2 in a priority queue may indeed correspond to the very same piece of content which is taking two different routes (i.e., different next-hop) to reach all the end-users.

When an intermediary node receives a message, two actions happen: first, based on the value of the message in the metadata, the message is inserted into the *sorted* priority queue; second, the encounter prediction technique embedded in CoHabit is called upon and the dynamic bundle is adjusted according to the Formula 6.1. Furthermore, to cater for the inevitable uncertainty that characterises human mobility, we require the underlying routing protocol to address and take into account the inaccuracy of encounter predictions. In particular, in scenarios where probabilities tend to under-estimate the actual encounter rate (i.e., causing the dynamic bundle to expand extensively), nodes must be able to step back and adopt a more cautious behaviour to avoid sending low-valued messages at the expense of the high-valued ones. CoHabit does not maintain information about how accurate its encounter predictions were; to cater for this, we thus simply rely on each node to maintain information about the proportion of messages it has already sent in the last Δt with respect to the $quota$. In CoHabit, if such proportion is above a given *Critical Boundary* threshold, the node ceases participation temporarily until the forwarded load during Δt falls below the Critical Boundary. Similarly, we introduce a *Loaded Boundary* threshold which works as follows: if the proportion of sent messages is below the loaded boundary, we let the bundle size n to grow as per formula 6.1; once the percentage of sent messages reaches the loaded boundary, we favour a more cautious behaviour by restricting the amount that the bundle size can expand. Once in the *Loaded Zone*, nodes are only allowed to expand the dynamic bundle up to $x \times quota$, with $x > 0$ and where smaller value of x presents more cautious behaviour by forwarding only the very high valued messages. The dynamic bundle behaviour during the loaded zone thus changes to Formula 6.2.

$$bundle.size = \begin{cases} queue.size(), \text{ if } queue.size() \leq quota \\ n \text{ otherwise} \end{cases} \tag{6.2}$$

with

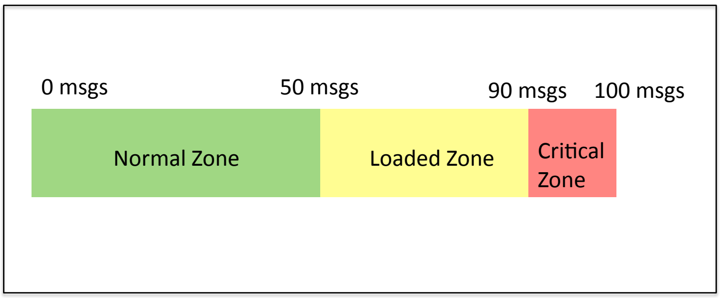


Figure 6.3: Behavioural Zones

$$n = \left\{ \#m_i | \sum_{i=1}^{queue.size()} Prob_{m_i} \leq quota \ and \ \ n \leq (x \times quota) \right\}$$

Let us illustrate this concept by means of an example. Figure 6.3 depicts the concept of CoHabit's zones, for an arbitrary node with maximum $quota$ of 100 messages in 5 days and the loaded boundary=50%. In this example, when the node has forwarded fewer than 50 messages in the past 5 days, it continues to schedule messages by adjusting the bundle size, that is, expanding the dynamic bundle so to include n number of messages (from the top of the queue) whose sum of their encountering probability is less than or equal to 50. Once the loaded zone has been reached, that is, more than 50 messages have been forwarded within the past 5 days, the node adapts a more cautious behaviour, so to only forward the most important messages. This is because its quota is fast dropping and the remaining resources should be devoted to really high valued messages. In adapting a more cautious behaviour, the node restricts the amount that the bundle can expand, limiting it to $x \times quota$. For instance, let us assume the arbitrary node in Figure 6.3 has forwarded 60 messages in the past 5 days (the remaining $quota$ =40), and variable $x = 1$; in this case, the node would schedule the messages from only the top 40 in the queue. Finally, once more than 90 messages have been forwarded, the node temporary takes a step back from exhausting its resources and stops forwarding messages until it has recovered from the critical zone, as in original CoHabit.

In the next section, we evaluate this realisation of our priority scheduling framework and demonstrate that our prioritisation technique maintains high delivery while also increasing user satisfaction significantly, by allocating available scarce resources into delivering most valued messages. We first present our simulation settings in Section 6.4, before reporting on the result of our extensive evaluation in Section 6.5.

6.4 Simulation Settings

In this section, we define our simulation settings, under which we have evaluated our proposed priority scheduling technique. Evaluations have been performed by means of simulation using OMNet++ [OMNeT++, 2010] network simulator.

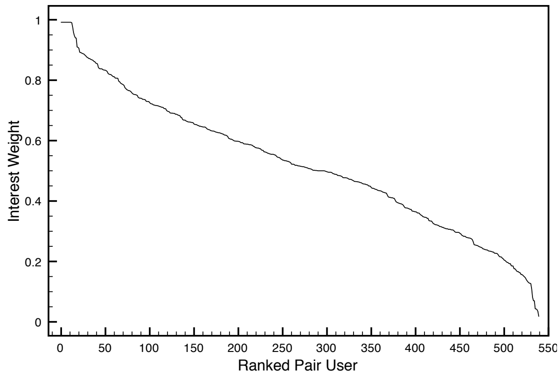
6.4.1 Datasets

To mimic a realistic settings for our evaluations, we required three types of datasets: one providing human mobility traces so to simulate encounters, another providing users social networks to determine who is interested in receiving content from whom and to what extent, and finally one to model publication rate.

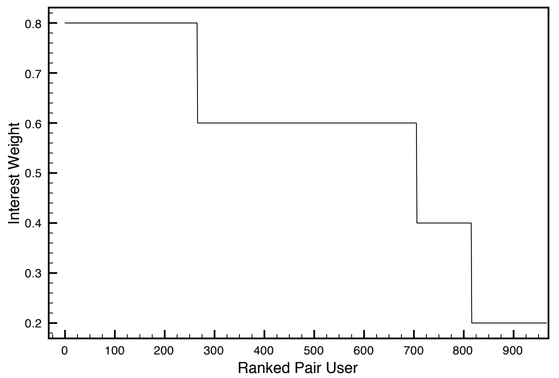
Mobility Traces: in terms of modelling user mobility, we experiment with two real mobility traces described previously in Chapter 5: Reality Mining and Cabs traces. The Reality Mining dataset is representative of a scenario with very few encounters (high inter-contact time); the Cabs traces are representative of an urban setting with more frequent connections instead. For Reality Mining, we have taken five months of colocation data (September to February); for Cabs, we have sampled movement data of 100 cabs for 21 days (for a more detailed description of these traces, refer to the page 70).

Social Network: for the purpose of evaluating our prioritisation scheduling framework, we require datasets presenting *weighted* social networks (the evaluation in this chapter is presented, without loss of generality, on the people-centric approach of modelling interest). To model who is interested in receiving content from whom, and to what extent, we have experimented with three distinct social networks, namely Last.fm, Advogato, and Reality Mining.

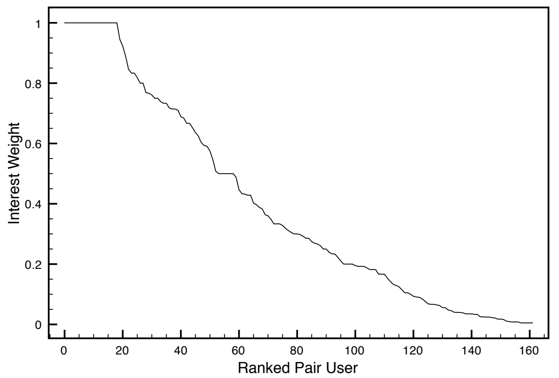
Last.fm [Last.fm, 2010] is a Web 2.0 music social networking website, where users explicitly declare who their social connections are. To sample this dataset, we first gathered 10,000 Last.fm users with a breadth-first search using the Audioscrobbler Web Service [Audioscrobbler, 2010]. We then sampled a (connected) sub-graph of 100 users; their in-degree distribution highlighted the long-tailed degree distribution typical of human social networks, as perviously illustrated in Figures 5.4. Note that social links have no weight in Last.fm; to obtain a *weighted* social graph, similarly to [Dell'Amico and Capra, 2008], we have computed, for each explicitly-declared link between users u_i and u_j, a weight $w \in [0, 1]$ as the cosine similarity between the vectors representing these users' top-50 most listened artists (that is, the more similar their musical tastes are, the higher the weight of the connection).



(a) Last.fm Social Network



(b) Advogato Social Network



(c) Reality Mining Social Network

Figure 6.4: Weight Distribution of the Selected Social Networks

Advogato [Advogato, 2010] is a community discussion board for free software developers; social links between developers are self-reported, and their weight can take one of the following discrete values: observer ($w = 0.2$), apprentice ($w = 0.4$), journeyer ($w = 0.6$), and master ($w = 0.8$). For our experiments, we extracted a sample of 100 users using breadth-first search, once again making sure to preserve the degree distribution of links amongst users.

Reality Mining provides information about voice calls and text messages exchanged between the participants in the study. We have used this information to extract an implicit social network whereby a link from user u_a to user u_b exists if u_a sent a text message or made a phone call to u_b; we have then associated a weight to each such link based on the normalised number of calls/texts user u_a had initiated. For instance, if u_a called u_b five times, u_c twice and u_d three times, then it would value contents from u_b, u_c and u_d as 0.5, 0.2 and 0.3 respectively.

To highlight the different properties of these datasets, Figure 6.4 plots the ranked distribution of social weights between each users' pair that exists in the social graph: as shown by the span of the $x\text{-}axis$, Reality Mining (Figure 6.4c) is by far the least connected network, with approximately 160 edges, as opposed to around 600 for Last.fm (Figure 6.4a) and 900 for Advogato (Figure 6.4b), for networks of equal size (≈ 100 users). The distribution of interest weights is shown on the $y\text{-}axis$: as expected, there exists a small subset of highly valued connections, and a much larger number of lesser valued ones; we expect our priority scheme to take advantage of these differences to forward the messages from the most valued connections first, thus increasing overall network satisfaction.

Publication Dataset: in terms of publications, we used the previously described Digg dataset (page 70, Section 5.4), where the rate of publication is set according to users activity in the content bookmarking website.

For completeness, we evaluated our work across all combinations of these social networks and mobility traces. Overlaying has been done multiple times at random, with the exception of the Reality Mining dataset, where there exists a direct (non random) mapping of users between movement and the inferred social graph.

6.4.2 Metrics

The goal of our prioritisation scheme is to guarantee enough resources are available to forward more important/valued messages, without compromising overall delivery, due to missed opportunities. In our experiments, we have thus computed the following *network-wide* metrics:

satisfaction gain computed as the difference between the average value of all messages delivered using priority scheduling on top of CoHabit, and using CoHabit (that is, a first-encounter/first-forwarded approach) alone.

delivery gain that is, the ratio of the number of messages delivered with and without priority scheduling on top of CoHabit.

6.4.3 Benchmarks

We compare satisfaction and delivery gain of our priority scheduling framework with respect to the original encounter-based CoHabit, which forwards messages based on the order of encounter, thus acting as an upper-bound in terms of delivery by not missing any opportunities, and lower-bound in terms of satisfaction, as it does not favour high-valued messages. We expect our priority scheduling to achieve higher satisfaction than its encounter-based variant, without compromising on delivery.

6.4.4 Parameters

As our realisation of the priority scheduling framework is built upon CoHabit, in here we list all the parameters and their values which will be used in the following evaluation (Table 6.1). Note that we have conducted a much broader set of experiments by manipulating both mobility and interest datasets; however, we only report results obtained using the following traces and parameters, as they are representative of our findings.

Network Parameters	
Simulation Duration	150 days for Reality Mining traces 42 days for Cabs traces
Number of nodes	96 for Reality Mining traces 100 for Cabs traces
Time-to-live	10 days for Reality Mining traces 3 days for Cabs
CoHabit Parameters	
Drainage threshold ($quota$)	50 message per 5 days
Critical Boundary	90%
Habit Parameters	
Regularity Interval	4 hour time slots
maxFS	10
maxHops	4

Table 6.1: Complete Parameter Settings

We now proceed to reporting the results of our conducted experiments based on the achieved delivery and satisfaction gain. In presenting the obtained results, we first focus on the effect that node's cautious behaviour has on prioritisation (Section 6.5.1), before presenting the results of our benchmark analysis based on extensive experiments on the complete overlay of all the datasets described above (Section 6.5.2).

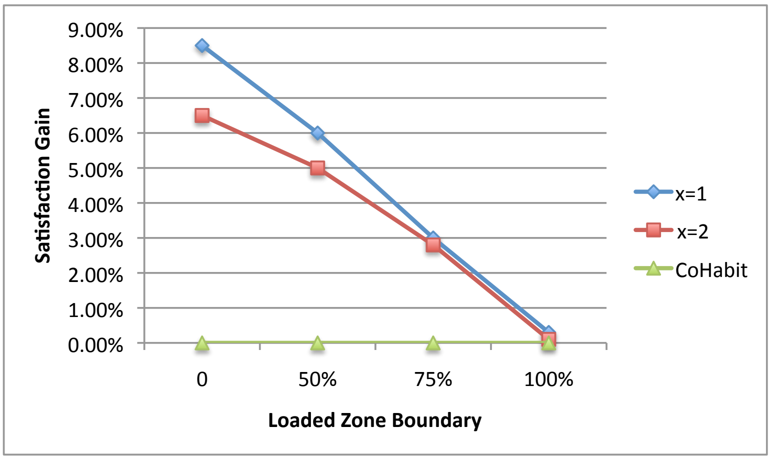
6.5 Results

6.5.1 Sensitivity Analysis

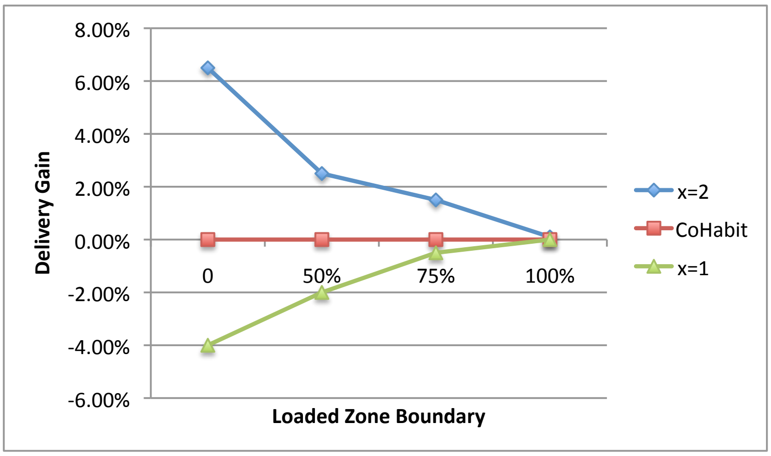
In this section, we investigate the effect of various variables on our prioritisation framework. In the first set of experiments, we investigate the effect that cautious behaviour of nodes has on the achieved delivery and satisfaction gain. As we previously described in Section 6.3, to cater for uncertainty of encounter predictions, we introduced the concept of loaded zone during which nodes limit their dynamic bundle to a restricted size of $x \times quota$. Figure 6.5 presents the effect of variable x and the loaded boundary for the Last.fm social network overlaid on top of the Reality Mining mobility traces, in comparison with our encounter-based benchmark protocol (i.e., original CoHabit without prioritisation).

Let us first examine the results in terms of our primary objective, that is, satisfaction gain. Figure 6.5a presents the achieved satisfaction (on y-$axis$) for various loaded zone boundaries presented on x-$axis$, where a loaded zone boundary of 0% means that the loaded zone is always set, and the bundle size is always restricted to $x \times quota$. Based on Figure 6.5a, the following two observations can be made: first, the lower the value of the loaded boundary, the higher the achieved satisfaction; this is because nodes start behaving cautiously sooner and thus only the most valued messages are forwarded at all the time. Secondly, the smaller the value of x, the higher satisfaction is achieved as only the top $quota$ messages are forwarded at all time (i.e., $x = 1$ and loaded zone boundary=0%).

We next investigate the *delivery gain* presented in Figure 6.5b. In terms of delivery, we can observe that the more flexible the degree of restriction (i.e., the bigger the value of x), the higher achieved delivery. Let us first focus on the results for when $x = 1$: in this case, as the cautious behaviour (i.e., loaded zone) is enforced sooner, the number of messages that are scheduled are restricted to $1 \times quota$, thus compromising the delivery by missing on encountering opportunities. On the contrary, for $x = 2$ (i.e., more flexible restriction), the achieved delivery is higher than the encounter-based CoHabit. To explain this gain (i.e., positive values on the y-$axis$), let us refer back to the description of the realisation of the priority scheduling framework at page 98, where the concept of different zones was presented. When a node is in the loaded zone, it slows down its forwarding rate by scheduling less messages, and thus avoids the tem-



(a) Satisfaction



(b) Gain in Delivery

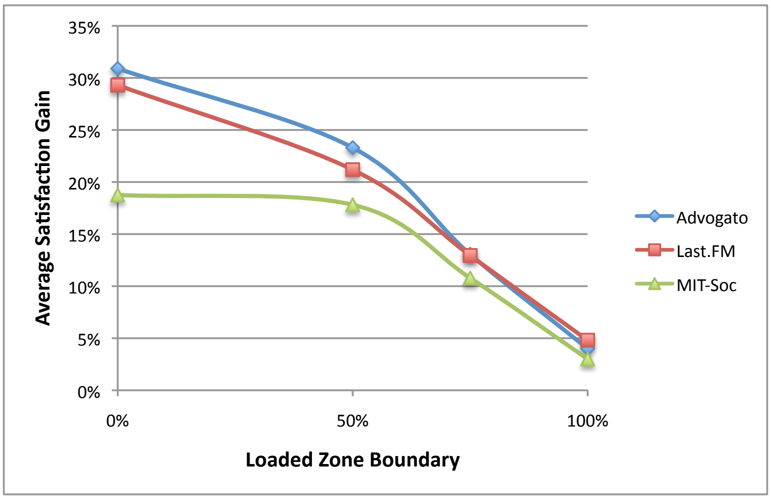
Figure 6.5: Effect of Cautious Behaviour

porary set back enforced when in critical zone. Therefore, our prioritisation scheme works best both in terms of delivery and satisfaction when nodes become cautious and slow down sooner (i.e., loaded zone boundary 0%), while allowing a higher degree of flexibility ($x = 2$).

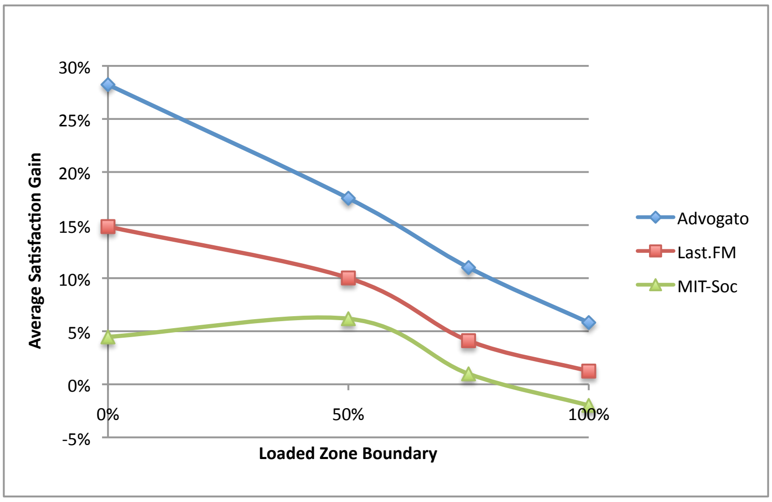
However, in the rest of evaluations we set variable x to its minimum value of 1; this is because the smaller values of x enables us to illustrate a more controversial case in which nodes' cautious behaviour can cause a trade-off between delivery and satisfaction to occur (as demonstrated in Figure 6.5).

6.5.2 Benchmark Analysis

In this section we report the results of the conducted benchmark analysis, highlighting the effectiveness of our prioritisation scheme in bringing higher overall network satisfaction, without compromising delivery, in comparison to the traditional encounter-based forwarding techniques. In so doing, we first investigate the results in terms of the *satisfaction gain* and then in



(a) Cabs Satisfaction



(b) Reality Mining Satisfaction

Figure 6.6: Satisfaction Gain

terms of *delivery gain*; finally, the *delay* caused by our priority scheme is measured with respect to CoHabit.

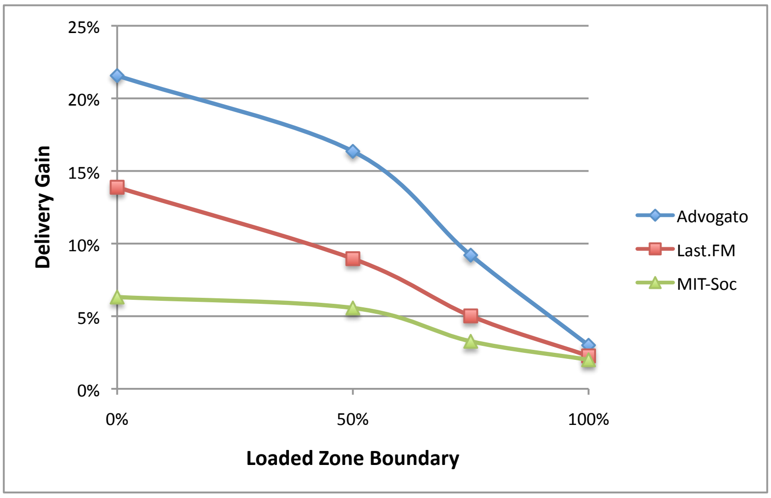
Satisfaction Gain. Figure 6.6 presents the satisfaction gain obtained for various overlays of social and mobility traces. More precisely, Figure 6.6a presents the results when using the Cabs mobility traces (and the three different social networks overlayed on top), while Figure 6.6b presents the results for the Reality Mining mobility traces (again with the three social networks overlayed on top). The following three observations can be made on those results:

- *Impact of Loaded Boundary* - as expected, the lower the value, the higher the satisfaction gain; this is because only $quota$ messages can be delivered at any Δt, and while CoHabit chooses to deliver based on a first-encountered/first-forwarded manner, our scheduler prioritises delivery of the $bundle\ size = quota$ most valued messages. With a boundary of 50, the gain is still very high (e.g., approximately 20% across all social networks for Cabs traces). However, for boundary of 100, the gain over CoHabit tends towards zero: this is because the bundle size can expand without any limit so that $bundle\ size \gg quota$, causing the encounters, and not message values, to drive the forwarding.

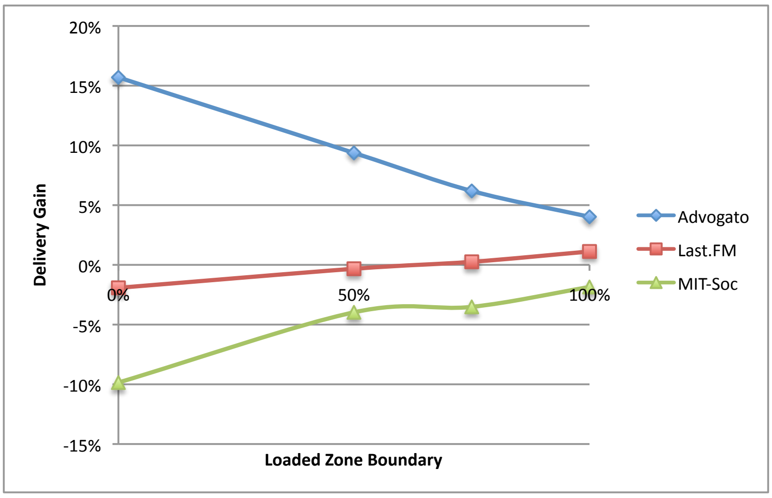
- *Impact of Mobility Traces* - for each value of loaded boundary and for each social network, the satisfaction gain is much higher for Cabs than for Reality Mining mobility traces. This is because the former has much more frequent encounters (lower inter-contact time): higher encounter probability means smaller bundle sizes and higher delivery of the most valued messages, thus pushing network satisfaction up.
- *Impact of Social Network* - finally, for each loaded boundary value and mobility traces, we observe that satisfaction gain is highest for the Advogato social network, followed by Last.fm and finally Reality Mining. The reason for this can be best explained by looking back at Figure 6.4: Advogato is the most connected social network, with approximately 25% of its social links having maximum weight. This means the priority scheduler can *choose* to first forward high valued messages in the queue, with neat gain over CoHabit. For Reality Mining social network, instead, connectivity in the social graph is much lower, and of lower values too; this means there is often little the scheduler can do in terms of prioritisation, hence lower gain over CoHabit.

As satisfaction gain has been computed as the average value of *delivered* messages. We now turn our attention to study the impact of priority scheduling on delivery rate, highlighting the cases where there is a trade-off between the two metrics.

Delivery Gain. Figure 6.7 depicts the gain in overall network delivery, while varying the loaded zone boundary parameter. In particular, Figure 6.7a presents results when using the Cabs mobility traces (and the three different social networks overlayed on top), while Figure 6.7b presents results for the Reality Mining mobility traces (again with the three social networks overlayed on top). The following observation can be made: delivery improves across all values of loaded boundary and social networks when deploying the Cab traces. A gain is also observed (though smaller) for the Advogato social network on top of Reality Mining mobility, while no gain nor loss is observed for Last.fm. Across



(a) Cabs Delivery



(b) Reality Mining Delivery

Figure 6.7: Delivery Gain

all these settings, when looking at both satisfaction and delivery, we can thus conclude that priority scheduling brings an unquestionable benefit, with very low values of loaded boundary being best for Cabs-like traces (small inter-contact time), and medium values for Reality Mining-like traces (high inter-contact time). The single setting where priority scheduling actually causes a loss in delivery is when using Reality Mining mobility traces and the inferred social network: a combination of high inter-contact time (i.e., few encounters, thus few forwarding), sparse social network, and cautious forwarding behaviour (boundary of 0) produces a 10% loss in delivery, with only a 5% gain in satisfaction (Figure 6.6b). Even a more aggressive behaviour (boundary of 100) does not really pay off, with both satisfaction and delivery being on a par with CoHabit.

We can thus conclude that, when opportunities for delivery are particular scarce, *and* the number of high valued message small, a fully opportunistic approach (first-encountered/first-delivered) is better suited than a priority one; on the other hand, in scenarios that give scope to priorisation (with more opportunities for delivery and/or higher differentiation in message values), our approach brings neat benefits, both in terms of satisfaction and delivery.

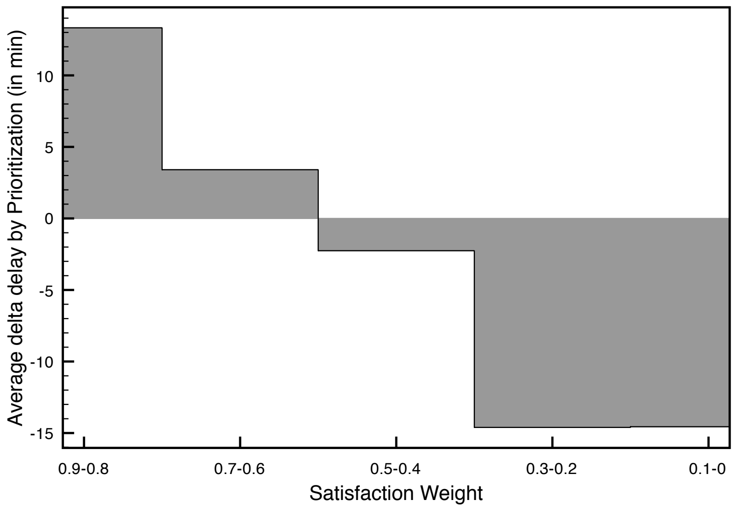


Figure 6.8: Delay Distribution

Delay. In order to investigate the effect of our prioritisation scheduling on delivery time of messages, we have conducted one last experiment whereby we measured the delivery time of those messages that both CoHabit and CoHabit with priority scheduling bring to destination. We have then computed the average gain (conversely, delay) in delivery time that priority scheduling brings for high value (conversely, low value) messages. Figure 6.8 presents this result for Cabs traces overlaid with Last.fm social network, where the x-$axis$ presents the satisfaction weight of the delivered messages from an end-user perspective, and the y-$axis$ presents the average *delta* delay that the prioritisation scheme exhibits. Similar results have been obtained for the rest of overlays of social networks.

As expected, high value messages ($w \in [0.8 - 0.9]$ on the x-$axis$) are delivered faster (positive values on the y-$axis$), when using prioritisation. This is done at the expense of low value messages ($w \in [0.1 - 0.3]$ on the x-$axis$), which are now delivered slower (negative values on the y-$axis$). Note that the gain (top-left corner) and loss (bottom-right corner) are of the same order of magnitude (on average approximately 13 to 15 minutes).

6.6 Conclusion

In this chapter, we presented a priority scheduling approach for participatory DTNs. We motivated this contribution by addressing the inadequacy of the state-of-the-art DTN routing protocols in reasoning on values of content the end-users. More precisely, common to all DTN routing protocols is the assumption that all messages are wanted equally in the network, and thus the forwarding is done based on the order of encountering the next hop carriers of such messages. However, different messages bring very different values to the end-users, as mobile devices have limited resources available, a prioritisation scheduling frameworks which can allocate the scarce available resources into forwarding the most valued messages is needed.

To bridge this gap in the literature, we have introduced a novel prioritisation scheduling framework, whereby messages are forwarded based on a combination of the likelihood of future encounters (physical layer) and the value that recipients attach to such messages (e.g., based on who produced the message). We have implemented this prioritisation scheme on top of CoHabit, and evaluated the gain that it entails in both end-user satisfaction and delivery, over a variety of real mobility traces and weighted social networks.

Chapter 7

Conclusion and Future Directions

In this chapter, we provide a summary and a critical evaluation of contributions that are presented in this thesis (Section 7.1). We further propose the future directions regarding content dissemination in participatory delay tolerant networks (Section 7.2), together with a longer term research vision (Section 7.3).

7.1 Summary of Contributions

This thesis was inspired by the need to close a gap in DTNs research which corresponds to users' participation. In particular, this thesis focused on content dissemination in human networks, where participation is driven by user interest as well as dictated by their device's available resources. By focusing on the participation dimension of such networks, we defined our research objective as designing a content dissemination network which can offer an effective and efficient delivery and an increase in user satisfaction, while distributing workload amongst participants fairly. We achieved this objective by means of the following contributions:

- We presented a source-based routing protocol that, by combining information from the physical and the application layer, delivers content to interested users (effectiveness) without overwhelming non-interested users as carriers (efficiency). We validated this protocol by comparing it against three of the state-of-the-art DTN protocols. Part of this contribution was presented at the IEEE International Symposium on the World of Wireless, Mobile and Multimedia Networks (WoWMoM 2009) [Mashhadi et al., 2009].

- We presented a folksonomy-based reasoning, which once applied to the content dissemination network, allows more interested destinations to be discovered, thus improving the effectiveness of the content delivery network. This research was presented at the ACM workshop on Challenged Networks (CHANTS 2010) [Lo Giusto et al., 2010].

- We introduced a load-balancing approach, which takes into account user participation from the point of resource constraints, and once integrated with a source-based DTN protocol, achieves fairness by distributing the workload amongst participants more uniformly. We validated this approach by comparing it against state-of-the-art DTN protocols. The outcome of this contribution is being published in the Elsevier Ad hoc Networks Journal [Mashhadi et al., 2011].

- We presented a priority scheduling approach for participatory DTNs, whereby messages are forwarded based on a combination of likelihood of future encounters and the value of content to the recipients. By relying on these two layers of information, our approach achieves a high satisfaction without compromising on effectiveness of the underlying DTN protocol. This contribution is currently under submission to the IEEE International Symposium on the World of Wireless, Mobile and Multimedia Networks Conference [Mashhadi and Capra, 2011].

7.1.1 Critical Evaluation

We evaluated the contributions of this thesis by means of simulation over a specific set of datasets. Therefore we expect the reported validation to hold only in scenarios where the network follows the same characteristics of those used in the tested datesets.

In particular, these are the environments where users mobility reflects daily interactions in an urban city. In such cases the colocations are short and the network suffers from frequent disconnection, as those exhibited in campus scenarios where students are colocated regularly for a short duration of time. However, unlike rural settings, there exist frequent encounters amongst nodes. This property of the mobility traces was observed in the two different mobility dataset that we used in evaluating this thesis: the Reality Mining traces and the San Francisco Cab traces. Moreover, our results and conclusions hold for scenarios where user interest in the produced content follows a long tail distribution, where most users are interested only in a small set of content, which defines their (selfish) involvement in the content delivery network. Such scenarios are representative of those observed in Web 2.0 applications, where the enormous amount of existing content contributes to this long tail distribution of users interest.

7.2 Future Work

To improve the performance of the techniques and protocols proposed in this thesis, the following future works are identified.

- Mobility prediction: while we rely on regularity predictions for building a view of the future colocation network so to be used by the source-based routing, we do not claim this prediction to be the most accurate. Indeed, in evaluating our proposed DTN protocol, we do not focus on the accuracy of this predication. Thus, this dimension of the protocol can be improved by a more sophisticated predication technique using the knowledge from the domain of artificial intelligence and by considering various factors such as time, environment and users' behaviour.

- Homogeneity in the network: this thesis solely focused on a homogeneous network, where we assumed all nodes to have the same amount of battery on their devices. Although this is often not the case in reality, we modelled the mobile network as a homogeneous network based on the following arguments: firstly, the focus of our work is on energy constrained handheld devices with Wi-Fi capabilities; this narrows the nodes in the network to modern mobile phones which nowadays have very similar battery resources. Secondly, as we previously mentioned in Chapter 5, we define the allocated battery as the amount of energy that can be fully dedicated to the application and which is not shared across other usage such as phone calls, etc. In other words, regardless of the heterogeneity amongst devices' energy resources, the amount of allocated battery to the particular content sharing application can still be assumed as uniform. Nonetheless, it is not realistic to assume that all users use their mobile devices the same way and thus consume their resources at a same rate. Therefore, it would be interesting to see how a more accurate energy model, reflective of the heterogeneity of devices, can improve fairness and participation amongst users in the network.

- Folksonomy matching: in this thesis, we introduced the first steps in tackling the problem of identifying interested users in produced content in the network. In so doing, we applied, a well-known web-based folksonomy matching technique, which relies on tag correlation, in a mobile setting. It remains open for future work, to compare and evaluate the precision of other folksonomy matching techniques. Note that this is not a trivial task, as an end-user study needs to be performed so to learn whether the content that was identified as interesting to an end-user was indeed of his interest.

7.3 Research Vision

This thesis offered various contributions with regard to content dissemination in delay tolerant networks concerning participation of users. This field could be further advanced in a number of ways, of which we have identified the following research streams.

- Hybrid Networks: in recent years, many metropolitan cities have seen successful development of fixed infrastructures (i.e., Hot Spots), providing users with a connection through Wi-Fi to the Internet for a given cost. In this regard, a content dissemination network which exploits both fixed and mobile nodes, so to deliver content *efficiently* is desired. In this thesis, we modelled this efficiency for such hybrid networks in terms of participation from the point of view of user interest, that is, we assumed that the fixed nodes are interested in every possible content and thus are favoured by our protocol as intermediaries (Chapter 4). However, given that these fixed nodes often have vast storage spaces, they can be further exploited in helping to reduce the content delivery time by proactively caching popular content, where the popularity of content is derived based on information from the application layer, concerning local community usage patterns and activities [Mashhadi, 2010]. Therefore, it would be interesting to explore how such hybrid settings can improve the state of the human content dissemination networks, by reducing the delay and network overhead.

- Content Centric Networks: in this thesis, we focused on facilitating content delivery in participatory delay tolerant networks. Indeed, in the context of DTNs, literature has mainly focused on how to discover delivery paths. However, an orthogonal issue is understanding *what content* is of relevance to users and thus should be routed via such paths. In addressing this problem, in this thesis we have taken the first steps into studying user/content matching in mobile networks by reasoning on distributed folksonomy-based matching [Lo Giusto et al., 2010]. As the magnitude of shared content increases, addressing this problem becomes more critical so as to avoid overwhelming users by masses of user-generated content.

  A new research stream is thus how to address the problem of helping users identify relevant content. In addressing this problem, a cross-layered approach which brings together research from the following fields can be exploited:

  - Social Networking: the social networks of users could be exploited in order to assist in identifying popular content. Much research has shown that the social affinities of users translate to their content consumption [Quercia et al., 2010, Golbeck, 2009]. The efficiency of content distribution in mobile networks can be considerably improved through the prediction of such patterns.
  - Collaborative Tagging: explicit information that is provided by user by means of descriptive tags, can facilitate content discovery [Heymann et al., 2008, Papagelis and Plexousakis, 2005].

  - Pervasive Computing: mobility information inferred from device status, such as its location as well as mobility pattern of the user can be exploited to offer users a more personalised content discovery experience [Rachuri et al., 2010, Mascolo, 2010, Kaasinen, 2003].

- Privacy and Maliciousness: finally, in this thesis, we did not focus on the issue of privacy, as we have considered it to be outside the scope of this research. We assumed that users are willing to share their interest profiles consisting of their social network, or their favourite tags, so to receive their desired content. However, while we focused on content dissemination networks and their participatory dimension, the issue of privacy for such networks remains an open research issue, specifically in the scenarios where users are not only selfish (as modelled in this thesis) but also malicious, and wish to harm the network and its users.

Bibliography

[Advogato, 2010] Advogato (Last Accessed December 2010). Advogato. http://advogato.org.

[Androutsellis-Theotokis and Spinellis, 2004] Androutsellis-Theotokis, S. and Spinellis, D. (2004). A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 36(4):335–371.

[Audioscrobbler, 2010] Audioscrobbler (Last Accessed December 2010). The music technology playground from last.fm. http://www.audioscrobbler.net.

[Bai et al., 2003] Bai, F., Sadagopan, N., and Helmy, A. (2003). The Important Framework for Analyzing the Impact of Mobility on Performance Of Routing protocols for Ad hoc Networks. *Ad Hoc Networks*, 1(4):383–403.

[Balasubramanian et al., 2007] Balasubramanian, A., Levine, B., and Venkataramani, A. (2007). DTN Routing as a Resource Allocation Problem. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 373–384, New York, NY, USA. ACM.

[Balasubramanian et al., 2009] Balasubramanian, N., Balasubramanian, A., and Venkataramani, A. (2009). Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications. In *Proceedings of the Ninth SIGCOMM Conference on Internet Measurement*, pages 280–293, New York, NY, USA. ACM.

[Balazinska and Castro, 2003] Balazinska, M. and Castro, P. (2003). Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. In *Proceedings of the First International Conference on Mobile Systems, Applications and Services*, pages 303–316, New York, NY, USA. ACM.

[Baldoni et al., 2005] Baldoni, R., Beraldi, R., Querzoni, L., Cugola, G., and Migliavacca, M. (2005). Content-based Routing in Highly Dynamic Mobile Ad Hoc Networks. *International Journal of Pervasive Computing and Communications*, 1(4):277–288.

[Bertier et al., 2009] Bertier, M., Guerraoui, R., Leroy, V., and Kermarrec, A.-M. (2009). Toward Personalized Query Expansion. In *Proceedings of the Second EuroSys Workshop on Social Network Systems*, pages 7–12, New York, NY, USA. ACM.

[Bettstetter et al., 2003] Bettstetter, C., Resta, G., and Santi, P. (2003). The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad hoc Networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269.

[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 *Proceedings of the International Conference on Wireless and Mobile Computing Networking and Communications*, pages 484–489, Washington, DC, USA. IEEE Computer Society.

[Boldrini et al., 2007] Boldrini, C., Conti, M., Jacopini, J., and Passarella, A. (2007). HiBOp: a History Based Routing Protocol for Opportunistic Networks. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–12, Washington, DC, USA. IEEE Computer Society.

[Boldrini et al., 2008] Boldrini, C., Conti, M., and Passarella, A. (2008). Context and Resource Awareness in Opportunistic Network Data Dissemination. In *Proceedings of the Ninth Annual International Symposium on the World of Wireless, Mobile and Multimedia Networks*, pages 1 –6, Washington, DC, USA. IEEE Computer Society.

[Broch et al., 1998] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. (1998). A Performance Comparison of Multi-hop Wireless Ad hoc Network Routing Protocols. In *Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking*, pages 85–97, New York, NY, USA. ACM.

[Burleigh et al., 2003] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H. (2003). Delay-tolerant Networking: An Approach to Interplanetary Internet. *IEEE Communications Magazine*, 41(6):128–136.

[Camp et al., 2002] Camp, T., Boleng, J., and Davies, V. (2002). A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications and Mobile Computing*, 2(5):483–502.

[Cao et al., 2002] Cao, Z., Wang, Z., and Zegura, E. (2002). Performance of Hashing-based Schemes for Internet Load Balancing. In *Proceedings of the Nineteenth Annual International Conference on Computer Communication*, pages 332–341, Washington, DC, USA. IEEE Computer Society.

[Carzaniga et al., 2000] Carzaniga, A., Rosenblum, D., and Wolf, A. (2000). Achieving Scalability and Expressiveness in an Internet-scale Event Notification Service. In *Proceedings of the Nineteenth Annual Symposium on Principles of Distributed Computing*, pages 219–227, New York, NY, USA. ACM.

[Cha et al., 2007] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y., and Moon, S. (2007). I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proceedings of the Seventh SIGCOMM Conference on Internet Measurement*, pages 1–14, New York, NY, USA. ACM.

[Chaintreau et al., 2007] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., and Scott, J. (2007). Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620.

[Cisco Visual Networking Index, 2010] Cisco Visual Networking Index (2010). Global Mobile Data Traffic Forecast Update for 2009-2014. White Paper.

[Clausen and Jacquet, 2003] Clausen, T. and Jacquet, P. (2003). Optimized Link State Routing Protocol (OLSR).

[CNET, 2010] CNET (Last Accessed December 2010). HTC Fuze: A Review. `http://reviews.cnet.com/smartphones/htc-fuze-at-t/4505-6452_7-33378478.html`.

[Cohen, 2003] Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *Proceedings of the First International Workshop on Economics of Peer-to-Peer systems*, pages 68–72, New York, NY, USA. ACM.

[Consolvo et al., 2005] Consolvo, S., Smith, I., Matthews, T., LaMarca, A., Tabert, J., and Powledge, P. (2005). Location Disclosure to Social Relations: Why, When, & What People Want to Share. In *Proceedings of the International Conference on Human Factors in Computing Systems*, pages 81–90, New York, NY, USA. ACM.

[Costa et al., 2008] Costa, P., Mascolo, C., and Musolesi, M. (June 2008). Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748–760.

[Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

[Cugola et al., 2002] Cugola, G., Di Nitto, E., and Fuggetta, A. (2002). The JEDI Event-based infrastructure and its application to the development of the OPSS WFMS. *IEEE Transactions on Software Engineering*, 27(9):827–850.

[Daly and Haahr, 2007] Daly, E. and Haahr, M. (2007). Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs. In *Proceedings of the Eighth International Symposium on Mobile Ad hoc Networking and Computing*, pages 32–40, New York, NY, USA. ACM.

[Das et al., 2000] Das, S., Castañeda, R., and Yan, J. (2000). Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks. *Mobile Networks and Applications*, 5(3):179–189.

[Dell'Amico and Capra, 2008] Dell'Amico, M. and Capra, L. (2008). SOFIA: Social Filtering for Robust Recommendations. *Trust Management II*, 263:135–150.

[Digg, 2010] Digg (Last Accessed December 2010). Content Bookmarking Website. `http://www.digg.com`.

[D'Souza and Jose, 2010] D'Souza, R. and Jose, J. (2010). Routing Approaches in Delay Tolerant Networks: A Survey. *International Journal of Computer Applications*, 1(17):9–15.

[Eagle and Pentland, 2006] Eagle, N. and Pentland, A. S. (2006). Reality Mining: Sensing Complex Social Systems. *Personal Ubiquitous Computing.*, 10(4):255–268.

[Gedik and Liu, 2005] Gedik, B. and Liu, L. (2005). Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of the Twenty Fifth International Conference on Distributed Computing Systems*, pages 620–629, Washington, DC, USA. IEEE Computer Society.

[Gedik and Liu, 2007] Gedik, B. and Liu, L. (2007). Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Transactions on Mobile Computing*, 7:1–18.

[Golbeck, 2009] Golbeck, J. (2009). Trust and Nuanced Profile Similarity in Online Social Networks. *ACM Transactions on the Web*, 3(4):12:1–12:33.

[Grundy and Radenkovic, 2010] Grundy, A. and Radenkovic, M. (2010). Decongesting Opportunistic Social-based Forwarding. In *Proceedings of the Seventh International Conference on Wireless On-demand Network Systems and Services*, pages 82–85, Washington, DC, USA. IEEE Computer Society.

[Guo and Keshav, 2007] Guo, S. and Keshav, S. (2007). Fair and Efficient Scheduling in Data Ferrying Networks. In *Proceedings of the Third International Conference on Emerging Networking Experiments and Technologies*, pages 1–12, New York, NY, USA. ACM.

[Hassanein and Zhou, 2003] Hassanein, H. and Zhou, A. (2003). Load-aware Destination-Controlled Routing for MANETs. *Computer Communications*, 26(14):1551–1559.

[Hei et al., 2007] Hei, X., Liang, C., Liang, J., Liu, Y., and Ross, K. (2007). A Measurement Study of a Large-Scale P2P IPTV System. *IEEE Transactions on Multimedia*, 9(8):1672–1687.

[Henderson et al., 2004] Henderson, T., Kotz, D., and Abyzov, I. (2004). The Changing Usage of a Mature Campus-wide Wireless Network. In *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking*, pages 187–201, New York, NY, USA. ACM.

[Heymann et al., 2008] Heymann, P., Koutrika, G., and Garcia-Molina, H. (2008). Can Social Bookmarking Improve Web Search? In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 195–206, New York, NY, USA. ACM.

[Hsu and Helmy, 2005] Hsu, W. J. and Helmy, A. (2005). IMPACT: Investigation of Mobile-user Patterns Across University Campuses using WLAN Trace Analysis. Technical report, Electrical Engineering Department, University of Southern California.

[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 SIGCOMM Workshop on Delay Tolerant Networking*, pages 244–251, New York, NY, USA. ACM.

[Ioannidis et al., 2009] Ioannidis, S., Chaintreau, A., and Massoulié, L. (2009). Distributing Content Updates Over a Mobile Social Network. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(1):44–47.

[Jain et al., 1984] Jain, R., Chiu, D., and Hawe, W. (1984). *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*. Eastern Research Laboratory, Digital Equipment Corporation.

[Jardosh et al., 2003] Jardosh, A., Belding-Royer, E. M., Almeroth, K. C., and Suri, S. (2003). Towards Realistic Mobility Models for Mobile Ad Hoc Networks. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking*, pages 217–229, New York, NY, USA. ACM.

[Johnson et al., 2002] Johnson, D., Maltz, D., Hu, Y., and Jetcheva, J. (2002). The Dynamic Source routing protocol for mobile ad hoc networks (DSR).

[Jun et al., 2006] Jun, H., Ammar, M., Corner, M., and Zegura, E. (2006). Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization. In *Proceedings of the SIGCOMM Workshop on Challenged Networks*, pages 245–252, New York, NY, USA. ACM.

[Kaasinen, 2003] Kaasinen, E. (2003). User needs for location-aware mobile services. *Personal Ubiquitous Computing*, 7(1):70–79.

[Kalman, 1960] Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of basic Engineering*, 82(1):35–45.

[Kim et al., 2006] Kim, M., Kotz, D., and Kim, S. (2006). Extracting a Mobility Model from Real User Traces. In *Proceedings of the Twenty Fifth International Conference on Computer Communications*, pages 1–13, Washington, DC, USA. IEEE Computer Society.

[Kleinberg et al., 2001] Kleinberg, J., Tardos, E., and Rabani, Y. (2001). Fairness in Routing and Load Balancing. *Journal of Computer and System Sciences*, 63(1):2 – 20.

[Ko and Vaidya, 2000] Ko, Y. and Vaidya, N. (2000). Location-Aided Routing (LAR) in Mobile Ad hoc Networks. *Wireless Networks*, 6(4):307–321.

[Last.fm, 2010] Last.fm (Last Accessed December 2010)). The World's Largest Online Music Catalogue. `http://www.last.fm`.

[Li and Wang, 2008] Li, F. and Wang, Y. (2008). Routing in Vehicular Ad Hoc Networks: A survey. *Vehicular Technology Magazine*, 2(2):12–22.

[Li et al., 2000] Li, J., Jannotti, J., De Couto, D., Karger, D., and Morris, R. (2000). A Scalable Location Service for Geographic Ad hoc Routing. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 120–130, New York, NY, USA. ACM.

[Li and Cuthbert, 2004] Li, X. and Cuthbert, L. (2004). A Reliable Node-disjoint Multipath Routing with Low Overhead in Wireless Ad Hoc Networks. In *Proceedings of the Seventh International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 230–233, New York, NY, USA. ACM.

[Liben-Nowell and Kleinberg, 2007] Liben-Nowell, D. and Kleinberg, J. (2007). The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031.

[Lindgren et al., 2006] Lindgren, A., Diot, C., and Scott, J. (2006). Impact of Communication Infrastructure on Forwarding in Pocket Switched Networks. In *Proceedings of the SIGCOMM Workshop on Challenged Networks*, pages 261–268, New York, NY, USA. ACM.

[Lindgren et al., 2004] Lindgren, A., Doria, A., and Schelén, O. (2004). Probabilistic Routing in Intermittently Connected Networks. *Mobile Computing and Communication Review*, 7(3):19–20.

[Lindgren et al., 2008] Lindgren, A., Mascolo, C., Lonergan, M., and McConnell, B. (2008). Seal-2-Seal: A Delay-Tolerant protocol for Contact Logging in Wildlife Monitoring Sensor Networks. In *Proceedins of the Fifth International Conference on Mobile Ad Hoc and Sensor Systems*, pages 321–327, Washington, DC, USA. IEEE Computer Society.

[Lo Giusto et al., 2010] Lo Giusto, G., Mashhadi, A. J., and Capra, L. (2010). Folksonomy-based reasoning for content dissemination in mobile settings. In *Proceedings of the Fifth SIGCOMM Workshop on Challenged Networks*, pages 39–46, New York, NY, USA. ACM.

[Mainwaring et al., 2002] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the First International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, New York, NY, USA. ACM.

[Manning et al., 2008] Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[Marina and Das, 2002] Marina, M. K. and Das, S. R. (2002). Ad Hoc On-Demand Multipath Distance Vector Routing. *SIGMOBILE Mobile Computing and Communications Review*, 6(3):92–93.

[Mascolo, 2010] Mascolo, C. (2010). The Power of Mobile Computing in a Social Era. *IEEE Internet Computing*, 14(6):76 –79.

[Mashhadi, 2010] Mashhadi, A. J. (2010). Proactive Caching for Hybrid Urban Mobile Networks. Research Note RN/10/05, UCL.

[Mashhadi and Capra, 2011] Mashhadi, A. J. and Capra, L. (2011). Priority Scheduling for Participatory Delay Tolerant Networks. Under Submission for the Twelfth Annual International Symposium on the World of Wireless, Mobile and Multimedia Networks.

[Mashhadi et al., 2009] Mashhadi, A. J., Mokhtar, S. B., and Capra, L. (2009). Habit: Leveraging Human Mobility and Social Network for Efficient Content Dissemination in Delay Tolerant Networks. In *Proceedings of the Tenth Annual International Symposium on the World of Wireless, Mobile and Multimedia Networks*, pages 1–6, Washington, DC, USA. IEEE Computer Society.

[Mashhadi et al., 2011] Mashhadi, A. J., Mokhtar, S. B., and Capra, L. (2011). Fair Content Dissemination in Participatory DTNs. *Special Issue on Ad hoc Networks Journal*. To appear.

[Mauve et al., 2002] Mauve, M., Widmer, A., and Hartenstein, H. (2002). A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39.

[McNamara, 2009] McNamara, L. (2009). *Context Aware Content Sharing in Urban Transport*. PhD thesis, University College London.

[Milo et al., 2007] Milo, T., Zur, T., and Verbin, E. (2007). Boosting Topic-Based Publish-Subscribe Systems with Dynamic Clustering. In *Proceedings of the International Conference on Management of Data*, pages 749–760, New York, NY, USA. ACM.

[Miluzzo et al., 2008] Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., and Campbell, A. T. (2008). Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the CenceMe Application. In

Proceedings of the Sixth Conference on Embedded Network Sensor Systems, pages 337–350, New York, NY, USA. ACM.

[MovieLens, 2010] MovieLens (Last Accessed December 2010). Movie Lens: A Movie Recommendation Website. `http://movielens.umn.edu`.

[Mtibaa et al., 2008] Mtibaa, A., Chaintreau, A., LeBrun, J., Oliver, E., Pietilainen, A., and Diot, C. (2008). Are You Moved by Your Social Network Application? In *Proceedings of the First Workshop on Online Social Networks*, pages 67–72, New York, NY, USA. ACM.

[Musolesi and Mascolo, 2006] Musolesi, M. and Mascolo, C. (2006). A Community Based Mobility Model for Ad Hoc Network Research. In *Proceedings of the Second International Workshop on Multi-hop Ad hoc Networks: From Theory to Reality*, pages 31–38, New York, NY, USA. ACM.

[Musolesi and Mascolo, 2008] Musolesi, M. and Mascolo, C. (2008). CAR: Context-aware Adaptive Routing for Delay-Tolerant Mobile Networks. *IEEE Transactions on Mobile Computing*, 8(2):246–260.

[Navas and Imielinski, 1997] Navas, J. and Imielinski, T. (1997). GeoCast: Geographic Addressing and Routing. In *Proceedings of the Third Annual International Conference on Mobile Computing and Networking*, pages 66–76, New York, NY, USA. ACM.

[OMNeT++, 2010] OMNeT++ (Last Accessed December 2010). A Network Simulation Framework. `http://www.omnetpp.org`.

[ONE, 2010] ONE (Last Accessed December 2010). The ONE: The Opportunistic Network Environment Simulator. `http://www.netlab.tkk.fi/tutkimus/dtn/theone`.

[Papagelis and Plexousakis, 2005] Papagelis, M. and Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering applications of artificial intelligence*, 18(7):781–789.

[Paradiso and Starner, 2005] Paradiso, J. A. and Starner, T. (2005). Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, 4(1):18–27.

[Perkins et al., 2003] Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc On-demand Distance Vector (AODV). *Request For Comments (RFC)*, 3561.

[Perkins and Bhagwat, 1994] Perkins, C. and Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *ACM SIGCOMM Computing and Communication Review*, 24(4):234–244.

[Piorkowski et al., 2009] Piorkowski, M., Sarafijanovoc-Djukic, N., and Grossglauser, M. (2009). A Parsimonious Model of Mobile Partitioned Networks with Clustering. In *Proceedings of the First International Conference on Communication Systems and Networks*, pages 1–10, Washington, DC, USA. IEEE Computer Society.

[Pogue, 2010] Pogue, D. (2010). Dont Worry about Who's Watching. *Nature: Scientific American*, 304(1):32.

[Pujol et al., 2009] Pujol, J., Toledo, A., and Rodriguez, P. (2009). Fair routing in delay tolerant networks. In *Proceedings of the Twenty Eighth International Conference on Computer Communications*, pages 837–845, Washington, DC, USA. IEEE Computer Society.

[Quercia et al., 2010] Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., and Crowcroft, J. (2010). Recommending Social Events from Mobile Phone Location Data. In *Proceedings of the International Conference on Data Mining*, pages 971–976, Washington, DC, USA. IEEE Computer Society.

[Rachuri et al., 2010] Rachuri, K., Musolesi, M., Mascolo, C., Rentfrow, P., Longworth, C., and Aucinas, A. (2010). EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research. In *Proceedings of the Twelfth International Conference on Ubiquitous Computing*, pages 281–290, New York, NY, USA. ACM.

[Radenkovic and Grundy, 2010] Radenkovic, M. and Grundy, A. (2010). Congestion Aware Data Dissemination in Social Opportunistic Networks. *SIGMOBILE Mobile Computing and Communication Review*, 14(3):31–33.

[Ramanathan et al., 2007] Ramanathan, R., Hansen, R., Basu, P., Rosales-Hain, R., and Krishnan, R. (2007). Prioritized Epidemic Routing for Opportunistic Networks. In *Proceedings of the First International MobiSys Workshop on Mobile Opportunistic Networking*, pages 62–66, New York, NY, USA. ACM.

[Rao et al., 2003] Rao, A., Lakshminarayanan, K., Surana, S., Karp, R., and Stoica, I. (2003). Load Balancing in Structured P2P Systems. In *Peer-to-Peer Systems II*, volume 2735 of *Lecture Notes in Computer Science*, pages 68–79. Springer.

[Rhee et al., 2008] Rhee, I., Shin, M., Hong, S., Lee, K., and Chong, S. (2008). On the Levy-Walk Nature of Human Mobility. In *Proceedings of the Twenty Seventh International Conference on Computer Communications*, pages 924–932, Washington, DC, USA. IEEE Computer Society.

[Rodrigues and Druschel, 2010] Rodrigues, R. and Druschel, P. (2010). Peer-to-Peer Systems. *ACM Communications Magazine*, 53(10):72–82.

[Rosenblum and Wolf, 1997] Rosenblum, D. and Wolf, A. (1997). A Design Framework for Internet-scale Event Observation and Notification. *SIGSOFT Software Engineering Notes*, 22(6):344–360.

[Sandulescu and Nadjm-Tehrani, 2008] Sandulescu, G. and Nadjm-Tehrani, S. (2008). Opportunistic DTN Routing with Window-Aware Adaptive Replication. In *Proceedings of the Fourth Asian Conference on Internet Engineering*, pages 103–112, New York, NY, USA. ACM.

[Scott and Bernardo, 2006] Scott, G. and Bernardo, A. H. (2006). Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208.

[Scott et al., 2009] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. (2009). Data Set Cambridge Haggle (v. 2009-05-29). `http://crawdad.cs.dartmouth.edu/cambridge/haggle`.

[Seligman et al., 2007] Seligman, M., Fall, K., and Mundur, P. (2007). Storage Routing for DTN Congestion Control. *Wireless Communication and Mobile Computing*, 7(10):1183–1196.

[Shorey and Miller, 2002] Shorey, R. and Miller, B. (2002). The Bluetooth Technology: Merits and Limitations. In *Proceedings of the International Conference on Personal Wireless Communications*, pages 80–84, Washington, DC, USA. IEEE Computer Society.

[Solis et al., 2010] Solis, J., Asokan, N., Kostiainen, K., Ginzboorg, P., and Ott, J. (2010). Controlling Resource Hogs in Mobile Delay-tolerant Networks. *Elsevier Computer Communications Journal*, 33(1):2–10.

[Sollazzo et al., 2007] Sollazzo, G., Musolesi, M., and Mascolo, C. (2007). TACO-DTN: a Time-Aware Content-based Dissemination System for Delay Tolerant Networks. In *Proceedings of the First International MobiSys Workshop on Mobile Opportunistic Networking*, pages 83–90, New York, NY, USA. ACM.

[Song et al., 2010] Song, C., Qu, Z., Blumm, N., and Barabasi, A. (2010). Limits of Predictability in Human Mobility. *Science*, 327(5968):1018.

[Spyropoulos et al., 2004] Spyropoulos, T., Psounis, K., and Raghavendra, C. (2004). Single-Copy Routing in Intermittently Connected Mobile Networks. In *Proceedings of the First Annual Conference on Sensor and Ad Hoc Communications and Networks*, pages 235–244, Washington, DC, USA. IEEE Computer Society.

[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 SIGCOMM Workshop on Delay Tolerant Networking*, pages 252–259, New York, NY, USA. ACM.

[Spyropoulos et al., 2006] Spyropoulos, T., Psounis, K., and Raghavendra, C. (2006). Performance Analysis of Mobility-Assisted Routing. In *Proceedings of the Seventh International Symposium on Mobile Ad Hoc Networking and Computing*, pages 49–60, New York, NY, USA. ACM.

[Tseng et al., 2002] Tseng, Y., Ni, S., Chen, Y., and Sheu, J. (2002). The Broadcast Storm Problem in a Mobile Ad Hoc Network. *Wireless Networks*, 8(2):153–167.

[Tuduce and Gross, 2005] Tuduce, C. and Gross, T. (2005). A Mobility Model Based on WLAN Traces and its Validation. In *Proceedings of the Twenty Forth Annual International Conference on Computer Communications*, pages 664–674, Washington, DC, USA. IEEE Computer Society.

[UCLU, 2010] UCLU (Last Accessed December 2010). UCL Union Website. `http://www.uclunion.org/clubs-societies/a-z.php`.

[Vahdat and Becker, 2000] Vahdat, A. and Becker, D. (2000). Epidemic Routing for Partially Connected Ad hoc Networks. Technical report, Duke University.

[Wang et al., 2007] Wang, W., Srinivasan, V., and Motani, M. (2007). Adaptive contact probing mechanisms for delay tolerant applications. In *Proceedings of the Thirteen Annual International Conference on Mobile Computing and Networking*, pages 230–241, Montreal, Quebec, Canada. ACM Press.

[Wi-Fi Alliance, 2010] Wi-Fi Alliance (2010). Wi-Fi CERTIFIED Wi-Fi Direct. White Paper: `http://www.wi-fi.org/news_articles.php?f=media_news&news_id=1011`.

[Wortham, 2009] Wortham, J. (2009). Customers Angered as iPhones Overload AT&T. *The New York Times*, 3.

[Ye et al., 2009] Ye, M., Tang, X., and Lee, D. L. (2009). Fair Delay Tolerant Mobile Data Ferrying. In *Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 182–191, Washington, DC, USA. IEEE Computer Society.

[Yin and Lin, 2005] Yin, S. and Lin, X. (2005). MALB: MANET Adaptive Load Balancing. In *Proceedings of the International Vehicular Technology Conference*, volume 4, pages 2843–2847, Washington, DC, USA. IEEE Computer Society.

[Yoneki et al., 2007a] Yoneki, E., Hui, P., Chan, S., and Crowcroft, J. (2007a). A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks. In *Proceedings of the Tenth Symposium on Modelling, Analysis, and Simulation of Wireless and Mobile Systems*, pages 225–234, New York, NY, USA. ACM.

[Yoneki et al., 2007b] Yoneki, E., Hui, P., and Crowcroft, J. (2007b). Visualizing Community Detection in Opportunistic Networks. In *Proceedings of the Second SIGCOMM Workshop on Challenged Networks*, pages 93–96, New York, NY, USA. ACM.

[Zanardi and Capra, 2008] Zanardi, V. and Capra, L. (2008). Social Ranking: Uncovering Relevant Content Using Tag-based Recommender Systems. In *Proceedings of the Conference on Recommender Systems*, pages 51–58, New York, NY, USA. ACM.

[Zhang et al., 2002] Zhang, L., Zhao, Z., Shu, Y., Wang, L., and Yang, O. (2002). Load Balancing of Multipath Source Routing in Ad Hoc Networks. In *Proceedings of the International Conference on Communications*, pages 3197–3201, Washington, DC, USA. IEEE Computer Society.

[Zhang et al., 2005] Zhang, X., Liu, J., Li, B., and Yum, Y. (2005). CoolStreaming/DONet: A Data-driven Overlay Network for Peer-To-Peer Live Media Streaming. In *Proceedings of the Twenty Forth Annual Joint Conference of the Computer and Communications Societies*, pages 102–111, Washington, DC, USA. IEEE Computer Society.

[Zhang, 2006] Zhang, Z. (2006). Routing in Intermittently Connected Mobile Ad hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys Tutorials*, 8(1):24–37.

[Ziegler, 2010] Ziegler, C. (Last Accessed: December 2010). Sprint falls in line, caps unlimited data at 5GB. `http://www.engadgetmobile.com/2008/05/19/sprint-falls-in-line-caps-unlimited-data-at-5gb`.