Collaboration in Opportunistic Networks

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Inform. Andreas Heinemann

geboren in Arolsen

Referenten
Prof. Dr. Max Mühlhäuser (TU Darmstadt)
Gerd Kortüm, Ph.D. (Lancaster University)

Tag der Einreichung:	30.04.2007
Tag der mündlichen Prüfung:	04.06.2007

Darmstadt 2007
Hochschulkennziffer D17

Acknowledgements

This work would not have been possible without the continuous support and encouragement of my colleagues and friends over the last years, which I would like to acknowledge here.

First and foremost I would like to thank my advisor, Max Mühlhäuser, for his faith in my work and for giving me excellent advice on many issues concerning this work and beyond. Next, I am grateful to Gerd Kortüm, for the fruitful discussions concerning this thesis and for acting as a second referee. In particular, his visit in 2005 was an inspiring experience for me.

I am grateful to all at Telecooperation and RBG for providing me with a friendly and supportive place to work. Special thanks go to Jussi Kangasharju for our regular meetings and his valuable advice, and to Tobias Straub for reviewing and discussing numerous aspects of this work with me.

It was a pleasure for me to co-author scientific publications with Erwin Aitenbichler, Andreas Görlach, Jussi Kangasharju, Fernando Lyardet, Max Mühlhäuser, Johannes Ranke, Tobias Straub, Wesley W. Terpstra, and Marco Voss.

Thanks to the "Austrians," Erwin Aitenbichler, Gerhard Austaller, and Andreas Hartl for their open-door-policy, whenever I ran into problems concerning administration and programming.

Many people improved this text with their reviews and comments. Thanks to Jussi Kangasharju, Gerd Kortüm, Max Mühlhäuser, Guido Rößling, and Tobias Straub. Thanks to Lara Schwarz for editing the final version.

Finally, thanks to my parents, sisters, and friends, especially Annette Ebert, for their mental support and patience during the course of this work.

Abstract

Motivation. With the increasing integration of wireless short-range communication technologies (Bluetooth, 802.11b WiFi) into mobile devices, novel applications for spontaneous communication, interaction and collaboration are possible. We distinguish between *active* and *passive* collaboration. The devices help users become aware of each other and stimulate face-to-face conversation (active collaboration). Also, autonomous device communication for sharing information without user interaction is possible, i.e., devices pass information to other devices in their vicinity (passive collaboration). Both, active and passive collaboration requires a user to specify what kind of information he offers and what kind of information he is interested in.

Object of Research: Opportunistic Networks. Spontaneous communication of mobile devices leads to so-called *opportunistic networks*, a new and promising evolution in mobile ad-hoc networking. They are formed by mobile devices which communicate with each other while users are in close proximity. There are two prominent characteristics present in opportunistic networks: 1) A user provides his *personal* device as a network node. 2) Users are a priori *unknown* to each other.

Objectives. Due to the fact that a user dedicates his personal device as a node to the opportunistic network and interacts with other users unknown to him, collaboration raises questions concerning two important human aspects: user privacy and incentives. The users' privacy is at risk, since passive collaboration applications may expose personal information about a user. Furthermore, some form of incentive is needed to encourage a user to share his personal device resources with others.

Both issues, user privacy and incentives, need to be taken into account in order to increase the user acceptability of opportunistic network applications. These aspects have not been addressed together with the technical tasks in prior opportunistic network research.

Scientific Contribution and Evaluation. This thesis investigates opportunistic networks in their entirety, i.e., our technical design decisions are appropriate for user privacy preservation and incentive schemes. In summary, the proposed concepts comprise system components, a node architecture, a system model and a simple *one-hop* communication paradigm for opportunistic network applications. One focus of this work is a profile-based data dissemination mechanism. A formal model for this mechanism will be presented. On top of that, we show how to preserve the privacy of a user by avoiding static and thus linkable data and an incentive scheme that is suitable for opportunistic network applications.

The evaluation of this work is twofold. We implemented two prototypes on off-the-shelf hardware to show the technical feasibility of our opportunistic network concepts. Also, the prototypes were used to carry out a number of runtime measurements. Then, we developed a novel two-step simulation method for opportunistic data dissemination. The simulation combines real world user traces with artificial user mobility models, in order to model user movements more realistically. We investigate our opportunistic data dissemination process under various settings, including different communication ranges and user behavior patterns. Our results depict, within the limits of our model and assumptions, a good performance of the data dissemination process.

Zusammenfassung

Motivation. Mobile Endgeräte sind zunehmend mit Technologien zur drahtlosen Vernetzung über kurze Distanz (bspw. Bluetooth, 802.11b WiFi) ausgestattet. Dies ermöglicht neuartige Formen der spontanen Kommunikation, Interaktion und Kollaboration. Hierbei wird zwischen *aktiver* und *passiver* Kollaboration unterschieden. Zum einen unterstützen Geräte in Kommunikationsreichweite die Nutzer dabei, sich als potentielle Partner wahrzunehmen und sich gegebenenfalls zu einem spontanen Gespräch (aktive Kollaboration) zusammenzufinden. Zum anderen können die Geräte autonom Informationen unter Nutzern verbreiten, sobald sich die Nutzer und somit die Geräte in Kommunikationsreichweite befinden (passive Kollaboration). Für die aktive wie passive Kollaboration teilt der Nutzer seinem Gerät mit, an welchen Informationen er interessiert ist bzw. welche Informationen er weitergeben möchte.

Forschungsgegenstand: Opportunistische Netzwerke. Durch die spontane Vernetzung mobiler Endgeräte formieren sich opportunistische Netzwerke (engl. *opportunistic networks*), die eine neue und vielversprechende Entwicklung auf dem Gebiet der mobilen ad-hoc Netzwerke darstellen. Opportunistische Netzwerke weisen zwei wesentliche Merkmale auf: 1) Ein Nutzer stellt sein *persönliches* Gerät partiell dem Netzwerk zur Verfügung. 2) A priori agiert ein Nutzer mit ihm *unbekannten* weiteren Teilnehmern des Netzwerkes.

Wissenschaftliche Fragestellung und Ziel. Der Einsatz von persönlichen Geräten und die Interaktion mit unbekannten Teilnehmern innerhalb eines opportunistischen Netzes werfen Fragen zum Schutz der Privatsphäre und zu Anreizen für die Nutzer auf. Anwendungen, die passive Kollaboration unterstützen, geben unter Umständen persönliche Informationen über einen Nutzer preis und gefährden so dessen Privatsphäre. Des Weiteren erfordern opportunistische Netzwerk-Anwendungen eine Möglichkeit, dem Nutzer einen Anreiz zu verschaffen, damit dieser sein persönliches Gerät partiell der Gemeinschaft zur Verfügung stellt.

Beide Belange sind zu betrachten, um hinreichende Akzeptanz von Anwendungen in opportunistischen Netzen zu erreichen. In vorangegangenen Forschungsarbeiten wurden der Schutz der Privatsphäre und Anreize für opportunistische Netzwerke – zwei wichtige Aspekte aus Sicht der Nutzer – nicht gemeinsam mit den technischen Fragestellungen untersucht.

Wissenschaftliche Beiträge der Arbeit und Evaluation. Als erster wissenschaftlicher Beitrag der Arbeit ist der gesamtheitliche Ansatz zu nennen: Die einzelnen technischen Entwurfsentscheidungen berücksichtigen den Schutz der Privatsphäre und sind geeignet, Anreizsysteme zu unterstützen. Diese Arbeit stellt hierzu passende Konzepte und Verfahren vor. Insbesondere konzipiert diese Arbeit Systemkomponenten, eine Netzwerkknotenarchitektur, ein Systemmodell und ein einfaches *one-hop* Kommunikationsparadigma für Anwendungen in opportunistischen Netzwerken und beschreibt deren Realisation sowie Evaluation. Hierbei liegt ein Schwerpunkt der Arbeit auf einem abstrakten Modell für profilbasierte Mechanismen zur Verbreitung von Informationen. Darauf aufbauend wird gezeigt, wie die Privatsphäre eines Nutzers mittels Verzicht auf statische Kommunikationsdaten geschützt werden kann. Des Weiteren stellen wir ein Anreizsystem vor, das sich als geeignet für opportunistische Netze erwiesen hat.

Die Evaluation gliedert sich in zwei Teile. Im ersten Teil werden zwei prototypisch realisierte Anwendungen auf Standard-Geräten vorgestellt. Die Prototypen dienen zum Nachweis der technischen Umsetzbarkeit der hier vorgestellten Konzepte und bilden die Plattform für eine Reihe von Laufzeitmessungen, deren Ergebnisse in dieser Arbeit vorgestellt werden. Der zweite Teil der Evaluation beruht auf einem Simulationsmodell für die Verbreitung von Informationen in opportunistischen Netzen. Dieses Simulationsmodell stellt einen eigenständigen originären wissenschaftlichen Beitrag dar. Es verbindet aufgezeichnete Daten aus der realen Welt, die einen Rückschluss auf die Nutzermobilität erlauben, mit Bewegungsmodellen. Ziel ist es, die Bewegung von Nutzern realitätsnah zu modellieren. Mit Hilfe einer Implementierung des Simulationsmodells untersucht diese Arbeit in verschiedenen Szenarien die Geschwindigkeit bei der Informationsverbreitung. Im Rahmen unserer Modellannahmen zeigt die Simulation eine gute Performanz bei der Verbreitung von Informationen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1991, Marc Weiser, at that time a researcher at Xerox PARC (Palo Alto Research Center), formulated his vision of a new area in computer science and called it *ubiquitous computing* [Wei91]. His vision promotes the idea of enabling people to move around and interact with computers more naturally than they currently do. Computers should become good, *invisible* tools. In his sense, an invisible tool is one that does not draw the user's attention towards itself. The user focuses on the task, not the tool. Weiser mentions eyeglasses as good tools. A user looks at the world, not the eyeglasses [Wei94]. Thus the tool disappears from the users' awareness [Wei93a]. In an ubiquitous computing environment, a user is

> *continually interacting with hundreds of nearby wirelessly interconnected computers* [Wei93b].

As a consequence, the idea of one or few *personal* computers per user has to be given up. Computers vanish into the background, "allowing people to just go about their lives" [Wei93b].

Another term that is closely related to ubiquitous computing is called *pervasive computing*. This term stresses more the idea of embedding computation power into the environment and thus being imperceptible as computers anymore. According to Lyytinen et al. [LY02], pervasive computing does not take node or user mobility into account. Figure 1.1 (adapted from [LY02]) relates the terms *mobile*, *pervasive* and *ubiquitous* computing to each other. Today, most people use pervasive computing as a synonym for ubiquitous computing.

The most prominent device that has conquered our everyday life and is basically *ubiquitously* available is the mobile phone, though is has not become an invisible tool in Weiser's sense. By the end of 2005, more than 810 million mobile phones were sold worldwide [Hei06]. In the top 5 economies in Europe (France, Germany, Italy, Spain, UK) on average 93,24 out of 100 inhabitants are subscribed to a mobile phone service [Int05].

Recently, more and more mobile phones (and other mobile devices, for example *Personal Digital Assistants* (PDAs) or laptops), are equipped with short

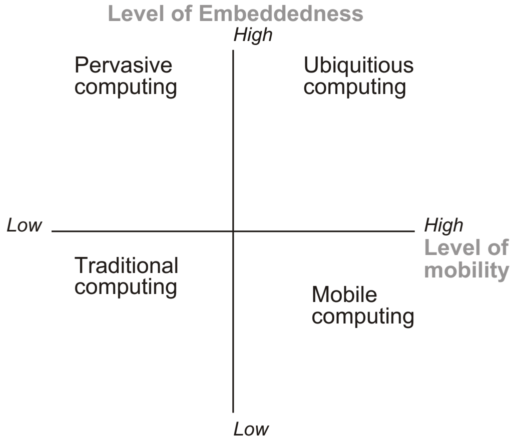


Figure 1.1: Traditional vs. pervasive vs. mobile vs. ubiquitous computing

range wireless communication capabilities (for Bluetooth module shipments in 2005 see [Blu05]). In most cases, either Bluetooth [Blu03] or 802.11b WiFi technology [IEE99] is integrated into the devices. The prevalent use of wireless connectivity is to synchronize personal data between a mobile device and a desktop computer (using Bluetooth) or have easy access to an institution's network (using 802.11 WiFi Wireless Access Points) and further to the Internet.

However, with the integration of short range wireless communication technology into mobile devices, a new network type called *opportunistic networks* and its corresponding applications based on *spontaneous interaction* and *collaboration* among devices and users is possible. We illustrate its capabilities by setting *word-of-mouth recommendation* among colleagues in contrast to adPASS, an opportunistic network application that disseminates advertisements among interested users.

> **Word-of-mouth recommendation**: Alice and Bob are co-workers sharing the same office. On her way to work, Alice passes a shop window that announces a digital camera: 20% off the regular price. Alice knows that Bob plans to buy such a camera and tells him about the advertisement when she arrives at her office. During lunch break, Bob visits the shop and buys the camera, glad that Alice has directed his attention to the offer.

Word-of-mouth recommendation is a well known and established way of communication and interaction among humans. There are two assumptions here: Alice and Bob know each other's interests in certain aspects of life and they meet on a regular basis (or know how to contact each other) to share information. In addition, on her way to work, Alice needs to be wide awake in order to notice the shop-window announcement.

adPASS, an opportunistic network application developed as part of this thesis mimics to some extent word-of-mouth recommendation.

> **adPASS**: Alice carries a mobile device with her. A personal profile, stored on her device, holds information about her interests and knowledge. The device is able to match her profile with other nearby devices by communicating wirelessly and without user interaction.
>
> A shop has put a fixed device next to the shop window. This device announces digital advertisements from the shop to passersby. As Alice passes the shop window, her device learns about the special offer for digital cameras.
>
> Alice physically carries the advertisement with her and passes it further to other users she encounters. All users interested in the ad (including herself and her colleague Bob) might take the chance and visit the shop in order to buy the advertised product.

adPASS differs from word-of-mouth recommendation in several ways: the users who exchange advertisements do not need to know each other. A match in their profiles is sufficient to share the ads. Next, a user does not need to keep his attention on the device. The device works without user interaction. Thus, Alice does not need to be wide awake in order to notice the ad.

In general terms, the following ideas are present in adPASS and other opportunistic network applications that aim at user collaboration:

1) User and device vicinity exploitation: An obvious requirement for short range communication is the *co-location* of users/devices at a certain time and place. This allows devices to pass information as illustrated by the adPASS example above.

Next, it raises the opportunity for users to meet *face-to-face* and make personal contact. In addition, to some extent, the usefulness of an application increases, since nearby users share the same physical context at a certain time and place. It is likely that these users share a common interest. Even if this is not true for every encounter, close vicinity allows getting to know new people or simply to share information.

2) Profile-based user interest expression: After two devices have discovered each other, there needs to be a way to determine if it is beneficial for a device and thus its owner to communicate further. This is achieved by employing a user profile on the device. A user profile expresses personal interest and knowledge. At the bottom line, a user wants to satisfy his interest and is committed to share his knowledge with other users. Therefore, an application needs a way to specify interest and knowledge and match interest against knowledge. This is a prerequisite for disseminating data.

3) Data dissemination: Whenever knowledge of a user Alice is able to satisfy interest of another user Bob by user profile matching, this knowledge is transferred from Alice to Bob. Given a number of users with the same interest, we observe a knowledge or data dissemination process. This process is additionally supported by user mobility: users physically carry knowledge while they move around.

4) Unpredictable communication pattern: Communication and information exchange takes place between mobile users that happen to be accidentally in communication range. In other words, a user can not rely on these kinds of applications to satisfy his interest. Therefore, questions like *"What is the menu at the university's cafeteria for today?"* or *"What are the opening hours for the city hall?"* are better answered by querying the Internet. Opportunistic networks simply offer a best effort functionality.

5) Open and unrelated user group: Apart from a few exceptions, most applications do not make any assumption about their participating users. Thus, in general, users are *unknown* to each other, act *independently*, and might also act *selfishly*.

Looking at these ideas as a whole, collaboration in opportunistic networks raises two central questions in terms of user acceptability. First, can we preserve the privacy of a user who uses adPASS or similar opportunistic network applications? Second, since a user makes its own personal device available to the opportunistic network, can we come up with an incentive scheme in order to stimulate the user's participation?

Privacy preservation and incentive schemes are two important *human aspects* present in opportunistic networks that have not been addressed together with the technical aspects in prior work. As we will see in the course of this thesis, these human aspects influence the technical tasks. Herein lies the novelty of this work. For example, our proposed *one-hop* communication paradigm is fundamental for an adequate privacy preservation. Figure 1.2 illustrates the interrelation between human aspects and technical tasks. We have addressed these topics within this work: *Algorithms and data modeling*, *communication* and to some extend *architectures* for opportunistic networks. Not touched are *resource management*, for example, how to cope with limited memory or battery supply since the advances in these areas make this less relevant, and *UI design*, being out of scope of this thesis.

Our results are of interest to all researchers working on opportunistic networks and related topics.

1.1 Objectives

The objectives of this thesis derive from the last section. One goal of this thesis is to formulate and define a system and communication model that is appropriate to integrate privacy preservation and an incentive scheme. Since part of this work is

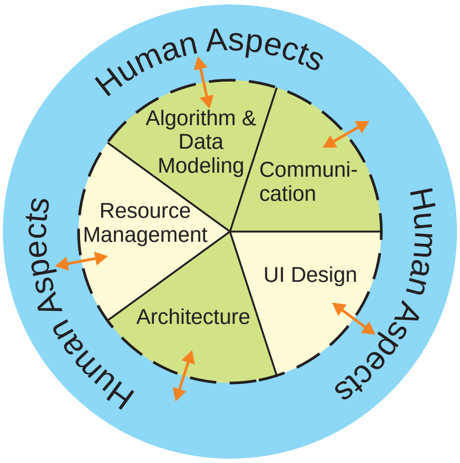


Figure 1.2: Human aspects and technical tasks in opportunistic networks

inspired by *word-of-mouth* or *gossip* like data dissemination, these ideas should be easily mapped onto our model.

Within the system model, a major aspect is the way data or information is expressed. A solution should not constrain itself to a certain technology or programming language and it should be simple and easy to understand.

Finally, the solutions should prove its *technical feasibility* by means of performance measurements and real-world tests using prototype realizations and its *effectiveness* should be validated by a data dissemination simulation.

1.2 Scientific Contribution

This thesis makes five contributions:

1) The first contribution is a system model for *opportunistic networks*. The model encompasses a communication model for data dissemination in opportunistic networks. It is based on a *one-hop* communication paradigm. In addition, the system model introduces two fundamental data structures, namely *iWish-list* and *iHave-list*, to allow users to express their information shares and needs to others. Within the model, nodes can be either mobile, i.e., users carrying a mobile device, or fixed. Fixed nodes are called *Information Sprinklers* and support *proximity based* services.

2) The second contribution is a formal model for describing *information* and *filter objects* that can be applied to the information. The formal model allows us to formulate programming language independent algorithms for matching user profiles based on iWish- and iHave-lists.

3) The third contribution addresses the human aspects in opportunistic networks; as said before: *user privacy preservation* and an *incentives scheme*. In order to preserve user privacy, a mechanism based on dynamic and user-self-generated aliases is described. An incentive scheme based on bonus points stimulates user participation in an opportunistic network. This contribution is aligned with our system and communication model.

4) The fourth contribution is the successful implementation of two opportunistic network prototype applications on off-the-shelf hardware. We conducted several real-world tests as well as application runtime measurements to evaluate the *technical feasibility* of the system model, the data dissemination mechanisms, and the incentive scheme.

5) The fifth contribution of this research is a novel *two-step* simulation model and simulator for opportunistic networks that combines real world user traces with artificial user mobility models. The simulator was used to evaluate the first and second contributions with respect to effectiveness on a broader scale and with different settings in respect to communication range and user behavior. By simulating the data dissemination process in an opportunistic network, the usefulness of the proposed system and communication model is shown.

1.3 Publications

Several aspects of this thesis have been published as research contributions in computer science conference proceedings or as a book chapter. In detail, emerging ideas and our opportunistic network concepts have been published in [HKLM03a, HKLM03b]. The design space and building blocks for opportunistic networks have been published in [HM05]. adPASS, a prototype that implements the incentive scheme presented in this thesis, is described in [SH04, HS03].

In addition, [VHM05] discusses a privacy preserving reputation system for opportunistic networks and [HRS04] looks into legal aspects according to the German law for adPASS and similar systems. Both topics go beyond the scope of this thesis.

1.4 Thesis Structure

This thesis is structured as follows. Chapter 2 presents related work for this research. First, we develop a number of conceptual and technical requirements for opportunistic networks that take human aspects into account. Then, for each requirement, prior work is presented. This includes the description of related projects that are similar to opportunistic networks. By analyzing these projects, we derive a number of common functionalities. In the last part of this chapter, we develop a design space for opportunistic networks that helps us to better categorize previous work.

Our opportunistic network concepts are presented in Chapter 3. First, basic definitions are given. This is followed by a system model description. Based on the *one-hop* communication paradigm, the data dissemination process is explained. Then, on a conceptual level, the data model and the notion of filters are introduced. The chapter concludes with a discussion on user acceptability, an outline of the proposed mechanisms to preserve user privacy, and the basic idea of our incentive scheme.

In Chapter 4, a formalization for the data modeling and profile matching task is developed. This model is used to outline language-independent algorithms and provides implementation guidance for important issues at design time. Some source code excerpts from the musicClouds prototype are given in order to show how to implement the model in the Java programming language.

User acceptability in opportunistic networks is addressed formally in Chapter 5. Our method to preserve user privacy and the incentive scheme is described in detail.

Chapter 6 evaluates the technical feasibility of our approach. A software architecture for opportunistic network nodes is presented first. This architecture was implemented within two prototype applications, adPASS and musicClouds, using off-the-shelf PDAs. The prototypes demonstrate the feasibility of this work. Feasibility is further confirmed by runtime-measurements and real-world tests.

Chapter 7 evaluates the effectiveness of the data dissemination process in an opportunistic network. We present our novel *two-step* simulation model and compare our approach with exiting work. Our simulation combines user traces from a real world experiment with artificial user mobility models. The simulator allows the data dissemination process to be tested with various parameters. For example, we conducted simulation runs with different device communication range and user behavior.

The thesis concludes with a summary of the major findings of this research and gives directions for future research in opportunistic networks.

Chapter 2

Background and Related Work

This chapter provides conceptual and technical background on the research issues of this thesis. It is divided into five sections. The first section briefly presents earlier work and recent research trends in opportunistic networks. Next, the second section defines a number of criteria that are essential for opportunistic networks that consider human aspects. Section 2.3 presents related research contributions for each criteria. We address each single criteria in turn (Section 2.3.3 to Section 2.3.6). Beforehand, opportunistic networks and Peer-to-Peer networks are discussed in Section 2.3.1, since there are a number of similarities. In addition, a brief overview on wireless short- to mid-range communication technologies is given in Section 2.3.2, since these technologies are fundamental for opportunistic networks.

Section 2.4 defines a number of building blocks for opportunistic networks. These building blocks are described as services and are integrated in the opportunistic network architecture (see Section 3.2.1). The building blocks allow an opportunistic network application developer to address human aspects if necessary. For example, the identity management service helps to preserve a user's privacy.

The second to last part, Section 2.5, develops the design space for opportunistic network applications. It defines two domains, *passive* and *active* collaboration. Passive collaboration focuses on pure device interaction and information dissemination without user interaction, whereas applications in the active collaboration domain help users discover each other and exploit a given physical user proximity to support the personal encounter of users. We conclude this chapter by summarizing our results.

2.1 Early Work and Recent Trends

This section briefly covers early work that exposes some ideas present in opportunistic networks and presents current research trends.

One of the ideas at the base of opportunistic networks is the short-range wireless communication of mobile devices carried by their users in order to make users aware of each other. The *Lovegety* [Iwa98] device is such a device. It helps introduce people to each other that happen to be in close proximity (approximately 5 meters). The device knows three different states. Whenever another device is found that is set to the same state, both devices beep and the holders may search for each other. Another example is the *Hummingbird* [HFW99], a mobile device to support mutual awareness between people who are in close vicinity of each other (approximately 100 meters). Being a custom-developed mobile device, it represents an early prototype of a so-called *inter-personal awareness device* (IPAD). Different to Lovegety, Hummingbird is designed for a closed group. It helps people notice the physical presence of other group members by playing a sound. Even if there is no visual or aural contact, a comforting *link* between users is created. This makes users more comfortable in unfamiliar settings like conference sites.

Newer projects use Bluetooth for mutual user awareness. We name just two: *BlueAware* and *BlueDating*. BlueAware was developed by Nathan Eagle as part of his Ph.D. thesis [Eag05, EP06]. It introduces people in close proximity to each other. For this, each user runs *BlueAware* on his Bluetooth-enabled mobile phone. BlueAware records unique Bluetooth identifiers from another device and submits this ID to the central *serendipity* server. The server uses the ID to query the database for a profile and matches this profile against the user's own profile. On a match (for example, high conformance in user interests), both users are notified, for example via a text message.

The second example, *BlueDating*, was developed by Beale et al. [Bea05]. It is very similar to BlueAware, however, it does not need a central server and works in a pure Peer-to-Peer fashion, i.e., the profiles are stored on the devices themselves and mutual profile matching is carried out on the devices as well.

Recently, the analysis of user traces gained interest among the opportunistic network research community. Chaintreau et al. [CHC+06] and Hui et al. [HCS+05] studied the transfer opportunities between mobile devices carried by humans. They found that the distribution of the inter-contact time of a pair of devices, i.e., the time gap between two successive contacts, approximately follows a power law distribution. Phanse and Nykvist [PN06] present a preliminary analysis of 2 user traces with a focus on statistical properties like node degree distribution and topological properties like cluster occurrences. An overview of opportunistic network research is given by Pelusi et al. [PPC06]. Most of the work focuses on opportunistic network message routing that assumes an end-to-end communication need between two or more communication partners, but without a direct path between them. This end-to-end communication need is not present if we assume an anonymous and unrelated user group. Thus, opportunistic network message routing is out of the scope of this thesis. In contrast, this work focuses on opportunistic data dissemination.

2.2 Opportunistic Network Criteria

As argued in the first chapter, current research does not look at opportunistic networks in its entirety. The human aspects privacy preservation and incentives are omitted in most prior work. A number of work has addressed various aspects in order to disseminate content in mobile ad-hoc communication settings. Since the term *opportunistic networks* is relatively new, various other terms are found in literature: *Peer-to-Peer networking* in combination with *mobile ad-hoc networking* [Dat03, DB04, GSX02, HW05, HDP03, KLW04, KLW03, LW05, LW02a], *en-passant communication* [GFH05], *spontaneous networking* [SP02, ANG02], *pocket switched networking* [HCG⁺05, HCS⁺05, SHCD06], and *mobile ad-hoc information system* [Kor02, KSP⁺01, KSP⁺01, KST99]. We will discuss these works in Section 2.3.

This section formulates an adequate number of criteria that are sufficient for opportunistic networks and its applications that deliberately take privacy preservation and incentives into account. A criterion is abbreviated by a capital letter for later reference, e.g., criterion **C** denotes *communication*. Some criteria are split into several aspects, whereas others are treated as a whole. Each criterion is based on the following assumptions:

1) Opportunistic networks are formed by individuals (carrying a mobile device) that are a priori *anonymous* to each other and have no relation.

2) Individuals make use of their *personal* devices. These devices may hold other personal data, for example, a calender or address book. In addition, device resources (battery power, memory capacity) are limited.

3) Wireless communication technology that is integrated in a device covers only a user's vicinity, i.e., at most a few hundred meters.

2.2.1 Communication

Opportunistic networks are formed by small devices that communicate over a wireless link with each other. These devices are either mobile, i.e., personal devices carried by a user, or fixed devices mounted at a dedicated location (see *Information Sprinkler* definition in Chapter 3.1). In this sense, opportunistic networks are closely related to *Mobile Ad Hoc Networks* (MANETs). We cite MANET characteristics from [CM99] below.

> **Mobile Ad-Hoc Network**: A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communications devices)[...] which are free to move about arbitrarily. [...] A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interfaces with a fixed network. [...] MANET nodes are equipped with wireless transmitters and receivers using antennas [...] At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multihop graph or "ad

> hoc" network exists between the nodes. This ad-hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

Obviously, MANETs are similar to opportunistic networks. Neither network type relies on a central component, for example, a central server; their architecture is decentralized by definition. Due to node mobility, nodes connect and disconnect since they move in and out of communication range. Connection and disconnection may also happen because devices are turned on or off unpredictably. Also, both networks may be formed by different kinds of mobile devices, such as a laptop, mobile phone, or PDA. These devices typically differ in battery duration, CPU power, and storage capacity. Communication in MANETs needs to provide the following functionalities:

- **Node discovery**: It has to be discovered if a node vanishes from a network (turned off or moved out of communication range) or if a node enters a network (turned on or moved in communication range).

- **Identity management**: Network entities, for example, nodes, users, or content, need to be identified. This functionality may include some form of privacy preservation, for instance by allowing users to act in an anonymous manner.

From a network stack viewpoint, MANETs reside on the network layer while opportunistic networks locate on the application layer and ask for very few network layer functionalities. In particular, an important difference between MANETs and opportunistic networks concerns routing. Routing allows end-to-end communication of network nodes via intermediates. Research in MANETs has put the focus on finding efficient routing algorithms that take both user mobility and limited node resources into account. The most prominent are *proactive* [PB94, JMC$^+$01] and *reactive* [JMB01, PR99] routing algorithms. Solutions that include geographical node positions are also common [Fre04, RT99]. Since MANETs have been investigated in the context of military networks, emergency response, and mobile sensor networks, all considered applications have several assumptions in common: all nodes are closely related to each other, trust each other, and share a common goal they want to accomplish.

Opportunistic networks, as we consider them, are formed between *anonymous groups of individuals*. This has an important impact on routing. Consider the situation in Figure 2.1 with A, B, and C as mobile nodes, in other words, individuals equipped with mobile devices. A is in communication range of B but not in communication range of C, who in turn is in communication range of B. If A wants to communicate with C, all traffic has to be routed via B. Bearing in mind that A, B, and C, a priori, do not know each other, the following questions arise:

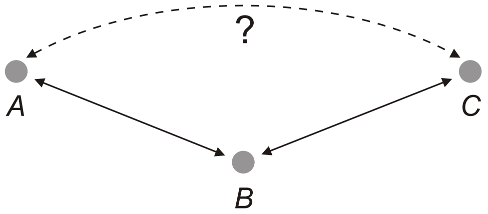


Figure 2.1: Multi-hop communication.

1. What is the incentive for node *B* to route messages between *A* and *C*? Why should node *B* be willing to donate part of its battery power to enable communication between *A* and *C*?

2. Why should node *A* and *C* trust and rely on node *B* for their communication? Node *B* could easily eavesdrop, manipulate, or simply reject messages.

These questions cannot be answered easily. Therefore, routing schemes for MANETs seem to fall short of providing for communication in opportunistic network settings. What is missing are incentives for users to forward messages and extra security mechanisms.

Therefore, the opportunistic network model proposes a wireless *one-hop* communication scheme where only directly connected nodes exchange messages. Directly connected nodes have a greater incentive to take part in the network, since they are able to satisfy their own information wishes. In addition, privacy preserving techniques are applicable, due to a *one-hop* communication paradigm (see Section 3.6.1). Criterion **C** is summarized in the box below:

Criterion C (Communication): Opportunistic networks have to provide means to discover close-by nodes and exchange messages (one-hop).

2.2.2 Data Dissemination

As stated in Chapter 1, the predominant idea for opportunistic network applications is to share information and knowledge with others in a spontaneous and ad-hoc manner. This has been selectively required by related work but not yet formulated clearly as a characteristic of a new type of network. Example applications based on sharing information include file sharing [HW05], target advertisements [SH04, SG02], tourist/event information sharing [SBB05, BBH02], gaming [GFH05], conference and spontaneous collaboration [SB03, Swi03], sharing cooperate knowledge [SP02], e-learning [ZNS03, ESN06], and the like. All these applications need to address the following tasks:

- A user needs to express his personal interest in a certain kind of information.
- Information to be shared within an application needs to be modeled appropriately to easily match personal interests.
- It needs to be possible to constrain information validity by time and location.

For example, a user needs to be able to express: *"I have interests in music events that take place in Darmstadt in July 2006"*. Or similarly, *"I have interest in music files from the artist 'Madonna' with a sample bit-rate* $\geq$ *192"*. Thus, we summarize criterion **D** as follows:

Criterion D (Data Dissemination): Opportunistic networks have to provide selective data dissemination means based on a general information model and expressive filter and constraints that consider time and location information.

2.2.3 Privacy

Depending on the application, criterion **D** raises privacy issues. If a user expresses interest in some kind of information or provides information or knowledge to other users/devices in the vicinity, there is a danger that other users exploit this information (we elaborate more on this issue in Section 3.6). Therefore, an application should offer means to preserve a user's privacy, for example, by allowing a user to stay anonymous within the network. Summarizing criterion **P**:

Criterion P (Privacy Preservation): Opportunistic networks have to provide means to preserve a user's privacy. Users may act under their identity, a pseudonym, or remain anonymous.

2.2.4 Incentive Scheme

Recall that an opportunistic network is formed by *personal* user devices. A priori, these devices serve other, strictly personal, purposes as well. A device may store calendar information, address lists, to-do lists, and the like. Since battery power is a limited and precious resource on such devices, an opportunistic network application should provide appropriate incentives for users that take part in a network, especially if a user just helps spreading information and has no further personal benefit from doing so. The adPASS application (see Section 6.1.1) serves as an example for integrating a bonus point based incentive scheme into an opportunistic network application. Criterion **I** is summarized below:

Criterion I (Incentive Scheme): Opportunistic networks have to provide means to reward participating users that help disseminate information.

2.2.5 Proximity Based Services

Device vicinity can not only be exploited by an application to make users aware of each other but also to implement a simple form of a *location-based service* (LBS) that we call *proximity based service*. A location-based service provides a mobile user with information that might be useful at the user's current or nearby location. For example, a user might want to find out the location of the nearest shopping center or gas station.

For a proximity based service deployed on an opportunistic network we have two options:

- Both the service provider and the service consumer are mobile. For example, a service provider is mounted on a public bus and the service consumer is a mobile user. Since the service provider is mobile, only very limited services are practical, for example, tourist information about the city or similar. The location resolution is rather coarse and not considered further in this thesis.

- The service provider is fixed (see Information Sprinkler definition in Chapter 3) and the service consumer is mobile. For example, a service provider mounted at a shopping mall is able to provide information like the shopping mall floor plan or special offers valid at shops at the mall. The location resolution is defined by the wireless communication technology in use. For Bluetooth Class 2 devices this is $\approx$ 10 meters. This kind of proximity based service is used by adPASS (see Section 6.1.1).

In comparison, current location-based services rely on a deployed infrastructure, e.g., a cellular network. These infrastructure premises make the LBS powerful, but also expensive. An infrastructure needs to be in place and service usage generates costs to a user. A location based service is able to answer queries including fine grained location information like *"My current position is Liebfrauenstr. 42, 64289 Darmstadt. I plan to go to Oberweg 12, 60318 Frankfurt in an hour. What gas stations are on my way?"*

Although location information is less accurate in proximity based services, it is favorable with respect to costs, since no infrastructure needs to be deployed in advance. A user's relative location is simply determined from the fact that his device is able to communicate with a nearby *fixed* node that knows its own location. Thus, *proximity*-based information like *"Hello user, you just passed a shop with your favorite red wine for 20% off"* is easily possible. In addition, since communication happens in an ad-hoc manner, no extra cost is generated for a user. We summarize criterion **L** as:

Criterion L (Proximity Based Service): Opportunistic networks have to provide means to exploit device vicinity to offer mobile users proximity based information and services.

2.3 Related Work

This section discusses related work for this research. We start with a comparison of opportunistic networks and mobile Peer-to-Peer networks, because of some prevalent similarities and subtle differences. This is followed by a brief overview on wireless short- to mid-range communication technologies that are convenient for opportunistic networks.

From Section 2.3.3 to Section 2.3.6, we review related work for each opportunistic network criterion, as formulated in the last section.

2.3.1 Opportunistic Networks and Mobile Peer-to-Peer Networks

Peer-to-Peer (P2P) networks have recently gained high interest in the research community. Looking at the definition for P2P as proposed by Schollmeier [Sch01], similarities to opportunistic networks appear.

> **Definition: Peer-to-Peer Network** A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers,...). These shared resources are necessary to provide the service and content offered by the network (e.g., file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource (service and content) providers as well as resource (service and content) requesters (Servent-concept).

The first thing that opportunistic networks and Peer-to-Peer (P2P) networks have in common is the integration of client and server functionality into one node or peer. An opportunistic network node consumes information and publishes information. Looking at the most prominent P2P application, file sharing on the Internet, a P2P node consumes files from other nodes that match a search query and allows other nodes to access locally stored files.

Thus, opportunistic networks fall under the definition of Peer-to-Peer network architectures. The definition above was given with Internet-based P2P applications in mind, as stated in the abstract of Schollmeier's work. Therefore, node mobility is not assumed in P2P networks. In addition, if we consider the Internet as the default P2P environment, a P2P network is several magnitudes larger than an opportunistic network.

Class	Power	Range
Class 1	100 mW	≈ 100 m
Class 2	2.5 mW	≈ 10 m
Class 3	1 mW	≈ 10 cm

Table 2.1: Bluetooth power classes

Since the main purpose of P2P networks is to share resources, discovery and sharing mechanisms, as well as identity management are present. Similar to MANETs and opportunistic networks, transient connectivity has to be handled by P2P networks, as most peers act autonomously and connect/disconnect unpredictably.

Identified and located resources are shared directly between two peers. However, in *pure* P2P networks, which do not rely on a central component, peers build a so-called overlay network for searching resources or content. This implies a cooperative behavior of individual peers and works well on the Internet, where online costs and peer energy consumption are not an issue.

For *mobile* P2P networks (MP2P), resource sharing without a benefit raises the same problems as mentioned before, namely *incentives* and *trust and reliability*. Within MP2P networks, the Peer-to-Peer concepts are mapped onto mobile networks. At the time of writing, there exists no coherent view of what is understood by mobile P2P. The only commonality is node mobility and therefore, nodes are equipped with wireless communication technology. Implementations range from *MP2P over mobile ad hoc networks* [Dat03] to *MP2P over cellular based networks* [HTA05b, HTA05a]. Application scenarios include pedestrians with mobile devices [HW05] or vehicles with wireless communication capabilities [XOW04].

2.3.2 Communication

This section gives a short introduction into the two most predominant wireless communication technologies available in the mass market, namely 802.11 WiFi [IEE99] and Bluetooth [Blu03]. We focus on 802.11 WiFi and Bluetooth since today these technologies are integrated in off-the-shelf mobile devices that are suitable for opportunistic networks. For example, most mobile phones are equipped with a Bluetooth module and most personal digital assistants (PDAs) are shipped with 802.11 WiFi modules.

Bluetooth: Bluetooth is a short-range wireless communication technology for forming wireless *personal area networks* (PAN) specified by the Bluetooth Special Interest Group (SIG), an industrial consortium established by Sony Ericsson, IBM, Intel, Toshiba and Nokia in 1999. Bluetooth is mainly used to connect devices, for example, personal digital assistants, mobile phones, laptops, or digital cameras, around a single person. Bluetooth operates on the 2.45 GHz frequency band. It distinguishes between three power classes (see Table 2.1) with different communication ranges. Bluetooth forms so-called *piconets*. A piconet consists of one *master* node and up to seven *slave* nodes. The specification allows two or more piconets to be connected together to form a so-called *scatternet*. Here some devices act as a bridge between two piconets, by playing the master role in one piconet and the slave role in another. The data rate starts from 723.1 kbit/s (Version 1.1 and 1.2) to 2.1 Mbit/s (Version 2.0).

According to opportunistic network applications with a focus on *active collaboration* (see Definition in Section 2.5), class 2 Bluetooth enabled devices are most suitable, since the communication range ($\approx$ 10 m) allows users to physically discover each other and switch to face-to-face collaboration.

802.11 WiFi: IEEE has created a family of specifications for wireless local area networks called 802.11. These specifications focus on the two lowest layers of the OSI model, the physical layer and the MAC (medium access) layer. 802.11 WiFi comprises several standards with different characteristics according to transmission speed and used frequency band, for example 802.11b (11 Mbps, 2.4 GHz) or 802.11g (54 Mbps, 2.4 GHz).

802.11b WiFi distinguishes between two types of networks, *independent networks* and *infrastructure networks*. An independent network is a pure ad-hoc network. Nodes in the network communicates directly with each other. An infrastructure network makes use of an *access point*. An access point is a fixed station, often connected to the Internet, that acts as a communication hub between any two devices. Thus, each packet from a node to another is relayed through the access point. This approach has two advantages. First, the wireless network coverage is extended. For two nodes to communicate, they do not need to be in communication range with each other, just in communication range with the access point. Second, an access point can help mobile nodes save power by buffering frames at the access point for the mobile node. The node itself stays in power-save mode most of the time and just wakes up to receive buffered frames if available.

Communication range differs between the specific standards. For example, 802.11b spans about 150 meters (outdoors) and 802.11g only 25 meters. Both ranges are suitable for opportunistic network applications that focus on *passive collaboration* applications (see Definition in Section 2.5), since no face-to-face user interaction is required by the application.

2.3.3 Data Dissemination

User profile-based data dissemination in opportunistic networks is closely related to *epidemic* algorithms for spreading information in distributed systems. These algorithms mimic the spread of a contagious disease and have been researched in the context of distributed data management (for example, see [DGH$^+$87]). In the same way as infected persons pass on a virus to those with whom they come into contact, each node in a distributed system passes new information to other randomly chosen peers. In turn, each of these nodes forwards the information to other randomly selected nodes, and so on.

Recently, a notable amount of research has addressed epidemic data dissemination in mobile ad-hoc networks. We will present the most prominent work now. A comparison and discussion follows (see page 21). Although the contributions vary in their details, we will see that the fundamental concepts are quite similar.

Datta et al. [DQA04] describe a selective information dissemination mechanism called *autonomous gossiping* (or A/G) for mobile, wireless connected mobile devices. Devices own a profile that expresses a user's information interest. A device profile is modeled as a set of fixed categories. This profile is advertised, i.e., broadcast locally to surrounding devices. In addition, each data item owns a profile. A profile for a data item is described as a tuple of its categories, its utility value, and its target location. A so-called *similarity* function is used for the replication and migration decision.

A data item tries to identify suitable hosts for migration or replication based on its own profile and the host's advertised profile. The underlying idea reflects an ecological and economic paradigm. Mobile hosts form habitats for the data items. The data items compete among themselves for limited resources, for example, device memory. The authors distinguish between four policies in A/G:

- Migration: A data item decides to move from one device to another device with higher hospitality.
- Replication: A data item with high utility decides to copy itself from one device to another to increase its population.
- Replica reconciliation: If a data item finds another copy of itself on a target device, only one data items stays there but its utility value is increased.
- Migration anyway: As an option, data items may store a geographical target within their profiles. Thus, a data item will migrate to all devices in the vicinity that move towards that location.

Datta et al. carried out some simple simulations to prove the usefulness of autonomous gossiping.

Görgen et al. [GFH05] describe an information dissemination protocol based on *single hop* communication between mobile devices. Devices form single hop Peer-to-Peer overlay networks according to interest in certain information categories. A simple quiz game application called *UbiQuiz* shows the feasibility of their communication scheme. In UbiQuiz, a user has to answer questions that are either stored on the device or received from other users' devices. The application aims to help students prepare for exams. New questions are collected in a software component called *InformationPool*. Questions and interest in questions are put in the *InformationGate*, another component that manages outgoing messages in a FIFO manner. UbiQuiz makes use of user profiles to express interest in certain question categories.

Goel et al. [GSX02] describe a protocol for Peer-to-Peer data dissemination in mobile ad-hoc networks. Their goal is to share popular data files, e.g., multimedia, among users carrying mobile devices. Their solution makes use of so-called tornado codes [BLMR98] to reduce network load. Using these codes, a mobile node is able to download coded file segments from different users at different times and locations and is able to re-construct the original file. Making use of a streets-and-buildings simulation model, they show that spreading a file is three times faster with tornado encoded file segments compared to splitting the file in segments.

Khelil et al. [KBTR02] investigate a model for information diffusion in MANETs. Inspired by the way an infectious disease spreads among individuals, a mobile node is either in state *susceptible* or in state *infective*. A susceptible node has interest in an information entity. An infective node has already received an information entity and passes this entity further to other susceptible nodes.

Becker et al. [BBH02] describe a system called *usenet-on-the-fly* for mobile phones that makes use of channels to share information in a mobile environment. The information spreading is limited by a hop count in the message. This has the disadvantage that an unlucky user might be one hop too far away from the information source, although he might be interested in receiving the information.

Hayes and Wilson [HW05] have developed an application to share music files (coded as MPEG Audio Layer 3) between Bluetooth enabled mobile phones. For this purpose, they adapted the Gnutella protocol [FP00] for Bluetooth usage. Since the user interface is fairly limited and the application should work without user attention, they propose an agent-based architecture. A search agent makes use of a user profile to query nodes in communication range for music files. The user profile is a simple list of keywords, for example `[Mozart, Beethoven]` would match any music file from these artists. Keywords are matched against ID3 tags [NM05]. Measurements showed that it took about 50 seconds to transfer a music file, of approximately 3 megabyte size, from one device to another.

Klemm et al. [KLW03, KLW04] propose a special-purpose approach for Peer-to-Peer file sharing on top of a mobile ad-hoc network called *Optimized Routing Independent Overlay Network (ORION)*. ORION creates an overlay network on top of a MANET that supports all kinds of messages required for file sharing, i.e, queries, answers, and file transmissions. The core idea is to set up overlay connections on demand, similar to reactive routing protocols like AODV [PR99] or DSR [JMB01]. This results in an overlay network topology that closely matches the underlying MANET topology. The authors compare their approach with the Gnutella [Cli01] protocol that makes use of TCP on top of a DSR-enabled MANET. Simulations show that ORION significantly increases search accuracy and reduces message overhead for searching.

Closely related to ORION is the work of Ding and Bhargava [DB04]. They propose and compare five routing protocols for Peer-to-Peer file sharing applications in mobile ad-hoc networks. They found that a cross-layer distributed hash table (DHT) protocol (for an introduction into DHT see [WGR05, GRW05]), which can process both network route and file requests, exposes the best routing complexity.

Lindemann and Waldhorst [LW02a, LW02b, LW05, Wal05] propose a distributed search service for mobile file sharing applications called *Passive Distributed Indexing (PDI)*. PDI uses local broadcast transmission of query and response messages. If a device cannot satisfy a query message, it retransmits the query message to adjacent devices. Query message forwarding is controlled by a *time-to-live* (TTL) value. Query results are cached at each device to reduce network load. Simulations show that PDI works well in mobile ad-hoc networks with a high node density. Here, TTL is set to 1. In setups with medium node density, TTL equals 2, i.e., 2-hop packet forwarding is applied. Entries in the index cache are replaced by a least-recently-used policy. PDI queries consist of keywords that are matched against a document. A document must match against all keywords (Boolean AND semantics). To evaluate a query, for each local document, a device stores a (*keyword*, *documentId*) tuple in its local index, where *documentId* consists of a pointer to the file in the local filesystem and a unique device identifier. PDI does not specify how a located document is transmitted between nodes. The authors rely on ad-hoc routing mechanisms or other means.

Scott et al. [SHCD06] investigate *pocket switched networks* (PSN) within the *Haggle* project [Int06]. A PSN uses mobile users' devices to build an opportunity-oriented network in order to transfer data between mobile devices. PSN aims to support three mechanisms by which data can be transferred, namely neighborhood connectivity between co-located devices, infrastructure connectivity to the Internet, and physical data transportation from place to place by exploiting user mobility. Currently, their research focuses on forwarding algorithms [CHC^{+}06] that make better use of human mobility. For this reason, the authors have conducted several real-world experiments to study data transfer opportunities between wireless devices carried by humans [HCS^{+}05]. PSN face several challenges: usability, naming, security, message forwarding, mobility, resource management [HCG^{+}05]. For security related issues, the authors name authentication, trust, reputation systems and incentives to cooperate as important topics. As Haggle is an ongoing research effort, the authors plan to address these issues in the future.

	Communication	Data Dissemination	Profiles
Datta et al. [Dat03, DQA04]	One-Hop	✓	✓
Görgen et al. [GFH05]	One-Hop	✓	✓
Goel et al. [GSX02]	One-Hop	✓	n/a
Khelil et al. [KBTR02]	One-Hop	✓	✗
Klemm et al. [KLW03, KLW04]	Multi-Hop	(✓)	(✓)
Lindemann and Waldhorst [LW02a, LW02b, LW05, Wal05]	One-Hop	✓	✓
Scott et al. [SHCD06, HCS+05, CHC+06]	Multi-Hop	n/a	n/a

✓= yes, ✗= no, n/a = not applicable

Table 2.2: Comparison of data dissemination approaches

Comparison: Most work relies on a one-hop communication scheme to support data dissemination in an ad-hoc network setting. The use of some kind of profile (node/user/data) to constrain data dissemination is also prominent. Table 2.2 summarizes the similarities of the discussed work.

An exception is the work of Goel et al., who do not give any information about usage of profiles. The model proposed by Khelil et al. does not consider different kinds of information and omits user profiles. The work of Klemm et al. is different in the sense that their file sharing protocol closely maps Peer-to-Peer Internet file sharing on ad-hoc networks. Implementing their protocol requires a user profile to store file queries, as well as to hold off users from focusing on their device while being on the move. In addition, a multi-hop approach assumes purely altruistic users. Since the Haggle project (Scott et al.) is still ongoing, nothing is said so far about a data dissemination mechanism or profiles. Currently, the Haggle project looks more into opportunistic message forwarding.

Remarkably, none of the discussed work considers user privacy preservation, although all authors consider a civilian setting, where users are unknown to each other and happen to meet accidentally, for example in a pedestrian zone. Additionally, it is obvious that incentive schemes are not considered as well. It is merely assumed that users are altruistic or have other reasons to share their private device resources. We strongly believe that this is a shortcoming. Our adPASS system closes this gap for a special purpose application, agreeing with Huang et al. [HCW04]: *“... incentive systems should be tailored to the needs of each individual application...”*, who have a down-to-earth view on incentive schemes.

2.3.4 Privacy Preserving Techniques

The emergence of ubiquitous computing technologies, with opportunistic networks being a part of it, raises user privacy issues. Especially the danger of tracking and monitoring user behavior in order to construct user profiles is present. In this sense, the success of Radio Frequency Identification (RFID) systems for automated object identification and supply chain applications has been criticized with respect to harming user privacy [WSRE03, KP04]. The storage of personal data on an RFID tag, as it is the case with E-passports, asks for specific measurements to preserve user privacy, for example *Basic Access Control* to ensure that data can be read only by authorized RFID readers [JMW05, SHR06].

Opportunistic network nodes are similar to RFID tags in the sense that they communicate with their surroundings without user interaction. They also store personal data and interests. Therefore, mechanisms for preserving user privacy are needed.

Most related work concerning privacy in ubiquitous computing addresses the protection of location data to obtain user *location* privacy (for a survey, see [GHT05, GHTM05]). We briefly present the most relevant work in the field now.

Snekkens [Sne01] presents concepts which may be useful when constructing tools to enable individuals to express a personal location privacy policy. Snekkens’ idea is that the individual should be able to adjust the accuracy of his location, identity, time, and speed and therefore have the power to enforce the need-to-know principle. The accuracy is dependent on the intended use of the data, and the use in turn is encoded within privacy policies.

Kong and Hong [KH03] describe their scheme ANDOR with the scenario of a battlefield in mind. ANDOR is a routing protocol addressing the problems of route anonymity and location privacy. The intention is that packets in the network can not be traced by any observing adversary. Additionally, their routing scheme provides unlinkability. Prior to one node’s ability to send a message to another, a route must be established through route discovery. This route discovery is achieved by broadcasting and forwarding packets. The sender of a message is anonymous, because it is impossible to judge whether a node is actually sending a message it generated or is simply forwarding a packet as part of a route.

Federrath et al. [FJP96] propose the application of mix networks (see also [Cha81]) in cellular networks like GSM, since in this kind of networks it is easy to track their mobile subscribers. In their system, the scheme does not keep the identity

– telephone number – of the recipient anonymous. Only the location of the recipient is protected. Remarkably, their system remains secure even if *all* intermediate nodes are observed by an adversary.

Beresford and Stajano [BS03] propose *mix zones* – an approach which is somewhat similar to mix networks. In these networks, the infrastructure provides an anonymity service. The infrastructure delays and reorders messages from subscribers within a mix zone to confuse an observer. One problem with this system is that there must be enough subscribers in the mix zone to provide an acceptable level of anonymity.

Gruteser and Grunwald [GG03] propose a mechanism called *cloaking* that conceals a user within a group of k people. They consider a user as *k-anonymous* if, and only if, he is indistinguishable from at least $k - 1$ other users. To achieve this, the accuracy of the disclosed location is reduced. Then, any of the people within the disclosed area could have been the particular user. Similarly, they consider reducing the accuracy of disclosure timestamps. Like Stajano and Beresford, they, too, measured anonymity in experimental setups, but unlike them Gruteser and Grundwald identified concrete values, which in their view provide a certain level of anonymity.

Comparison: With respect to user privacy preservation in opportunistic networks, none of the above mechanisms are suitable. This is due to the fact that privacy preserving mechanisms are tailored to the considered applications. However, all approaches teach a fundamental lesson: in order to preserve user privacy, the source, i.e., the user's identity, of an event or information has to be obfuscated from an observer. Thus, in order to preserve user privacy within an opportunistic network setting, this thesis proposes to avoid a priori any static data or information that could later be linked to a particular user. This is elaborated on in Section 5.1.

2.3.5 Incentive Schemes

Incentive schemes are vitally important to (mobile) Peer-to-Peer networks or mobile ad-hoc networks that are formed by unrelated and selfishly acting nodes, often called *free-riders* [AHrg]. For example, Saroiu et al. [SGG03] showed that only 7% of clients in the Peer-to-Peer Gnutella network share more than 1000 files. On the other hand, 25% of its users do not share any files and about 75% of the clients share 100 files or less.[1] Since opportunistic networks are related to (mobile) Peer-to-Peer networks and mobile ad-hoc networks, we discuss incentive schemes in these areas.

[1] These values are accumulated, thus, if 7% of clients share more than 1000 files, 93% share less than 1000 files.

Golle et al. [GLBML01] have addressed the incentive issue in centralized Peer-to-Peer networks. They propose and analyze several micro-payment mechanisms to encourage file sharing.

Crowcroft et al. [CGKÖ04] propose a pricing mechanism for mobile ad-hoc network nodes as an incentive to forward network packages. Each user has a credit balance and receives an initial endowment when he joins the network. The node balance is increased by forwarding traffic to other users and decreased based on the cost of forwarding the traffic to its destination.

Mannak et al. [MdRK04] have conducted a small study on users' motivation and decision to share resources in Peer-to-Peer networks. They found out that 50% of the questioned users would share more, if some materialistic incentives, for example earning money, would be dispensed by the application. Herein lies the motivation for coupon based systems like adPASS [SH04].

Ratsimor et al. [RFJY03] describe a system similar to adPASS. It is called *eNcentive* and allows mobile agents to spread digital advertisements with embedded coupons among mobile users in a Peer-to-Peer manner. Their agent based framework runs on both mobile devices and advertisers' portals. A portal is a fixed station and takes the role of an Information Sprinkler. Ratsimor et al. propose two discount reward models. Model A uses $f(x) = (1/1 + e^{\sqrt{x}}) \cdot 0.3$ with x being the amount of successful promotions. Model B follows a *threshold* reward model. The first ten users get 5% off, for the ten to thirty successful promotions, users get 10% off, and above that, users get 20% off. Discount is only granted on successful promotions. In contrast to our proposed bonus point model (see Section 5.2), a user cannot affect his chance of being rewarded, for example, by choosing a different strategy.

Garyfalos and Almeroth [GA04, AG04] describe *Coupons*, an incentive scheme that is inspired by the *eNcentive* framework and prior publications of the author of this thesis [HKLM03a, HKLM03b]. *Coupons* gives users credit for forwarding information to other users in an ad-hoc network. By simulating, they show that it is possible to achieve a good information spreading rate by employing less greedy and aggressive user behavior, i.e., users do not take every message and do not re-broadcast every message. This leads to an overall reduction of network messages by 90%. Contrary to adPASS, users cannot affect their chance of being rewarded at all. A message or coupon has a fixed number of empty slots (they use 5 slots in their simulation). Whenever a user receives a message, he fills an empty slot with his ID. This allows the user to claim a reward later. Mapped to the bonus point model, this means that the total number of bonus points is always fixed and a user may take one point per message.

Comparison: Incentive schemes have gained some interest in Peer-to-Peer file sharing applications in order to remedy the *free-riders* problem. Garyfalos and Almeroth and the work of Ratsimor et al. are closely related to our incentive scheme.

The largest difference between related incentive schemes and this work is the omission of privacy preservation in prior work. Herein lies one novelty of this work. For the first time, privacy preservation was considered together with an incentive scheme.

2.3.6 Proximity Based Services

Location Based Services (LBS) have been widely researched and are available as or integrated in commercial products. See *D'Roza* and *Bilchev* [DB03] for an overview of technologies and standards available. For location based services to work, first, the user location has to be determined. The most prominent technologies are either GPS based or based on GSM cellular location. Both approaches bear some disadvantages. GPS works only outdoors and on top of the raw location data, all service provision has to be done by other means, for example by querying a local database. GSM-based approaches allow for service delegation and composition somewhere in the infrastructure or back-end system. While this allows for greater flexibility and up-to-date data delivery, this approach usually generates costs for the user and since the location data is determined by the infrastructure, location privacy is at stake. On the other hand, as *Rao* and *Minakakis* put it *"LBS can be a new source of revenue opportunity for multiple stakeholders in the mobile value chain."* [RM03].

Opportunistic networks allow for a much simpler, more decentralized possibility. We call this approach *proximity based service*, since the accuracy is less than with LBS. We assume that a fixed station knows its location and broadcasts information that is valid and relevant for this location. A mobile device that moves into communication range with a fixed station is automatically co-located with that station. Thus, there is no need to determine the device location by other means in order to offer a service.

Kaasinen [Kaa03] conducted a study on user needs for location based services from the user's point of view. Encouraging for this thesis is their finding that users would appreciate a service that pushed information onto their devices, as long as the information is useful. Especially the attitude towards location based advertisements is positive, as long as the user has the ability to select what kind of advertisements they receive. This motivates the usage of user profiles as proposed for opportunistic network applications. In addition, Kaasinen demands for an LBS: *"The user should be allowed to remain anonymous when (s)he wants"*.

The design of adPASS took these facts into account. adPASS is one example of such an proximity based service. We will discuss other work in that field now.

Ojala et al. [OKA$^+$03] describe the *SmartRotuaari service system*, a service environment for context-aware mobile multimedia services, deployed in Oulu, Finland. The system offers a variety of services to users that are provided with WiFi enabled PDAs. These include map-based guidance, personal communication with friends, personalized news, mobile payment and mobile advertising. User location is derived by determining proximity to a client-side pre-registered WiFi wireless access points or by a commercial module that exploits WiFi signal strength of the client.

Aalto et al. [AGKO04] describe a location based mobile advertising system based on Bluetooth proximity and WAP [Ope06]. A Bluetooth sensor, mounted behind a shopping window, detects Bluetooth enabled mobile phones by a unique ID. This ID, together with location information, is sent to an advertisement server. The server maps the ID to a user and checks if there are advertisements waiting for delivery at the location. If yes, the new advertisements are pushed onto the users' mobile phones using WAP Push.

Kurkovsky and Harihar [KH06] developed the *SMMART* prototype. SMMART, an abbreviation for *System for Mobile Marketing: Adaptive, PeRsonalized and Targeted*, allows for the delivery of targeted advertisements to a user's mobile device. Their system uses fixed 802.11 WiFi for communication. Fixed nodes located at dedicated places like shopping malls deliver advertisements to PDAs of passersby. In order to receive only desired information, a user specifies his interest via a list of keywords, for example, 'Cranberries', 'Dire Straits', and 'Police' to express interest in these musicians. These keywords are submitted to the advertising node and matched against offers. In addition, the system proposes new keywords to the mobile node. For example, the keyword 'Sting' might be proposed to the user, since the singer was a member of the group 'Police' and a user might be interested in solo albums of 'Sting' as well. Also, the system offers related product advertisements like DVDs or books about related topics.

Rudström et al. [RSCH04] describe *MobiTip*, a system that allows its users to express their opinions on anything of interest in the environment. Opinions are aggregated and presented to the users as tips or recommendations. Opinions are entered in free text form on the user's device (a mobile phone) and shared in a Peer-to-Peer manner on-the-fly with users nearby using Bluetooth. A typical example is a shopping mall, where MobiTip users share their personal views on certain shops or product offers. The core MobiTip system can be extended by so-called *connection hotspots*. A connection hotspot is placed at a selected location, e.g., the entrance of a shopping mall, to collect tips and pass them to future visitors.

Comparison: The idea of offering an information service based solely on proximity is present in all cited work. Nonetheless, there are differences and shortcomings in prior work, which we are going to discuss now.

The *SmartRotuaari service system* is simpler than adPASS, since their mobile advertising does not allow users to express likes or dislikes in certain advertisements. Thus, no filter capabilities are available for the user.

Similarly, the work by Aalto et al. does not support individual user profiles, i.e., users cannot specify what type of advertisement they are interested in or not. Also, neither privacy nor security issues, like encryption of network messages, have been considered in the system design.

SMMART is similar to adPASS as well. However, SMMART does not allow users to pass advertisements to other users they encounter outside and away from the advertising node. Thus, advertisements are not spread widely. Also, SMMART does not implement any incentive scheme.

From a privacy perspective, Kurkovsky and Harihar claim that SMMART guarantees a high level of privacy due to the fact that the PDA does not communicate any personally identifiable information, just keywords. This is true and also holds for adPASS. But, the authors overlook that devices are identifiable by their unique MAC address and profiling is possible, once the MAC address is linked to a name, for example via a credit card based payment of the advertised product. This is precisely the reason why this work advocates the usage of changing MAC and IP addresses, as well as self-generated keys as user aliases (see Section 5.1).

2.4 Building Blocks for Opportunistic Networks

The human aspects privacy preservation and incentives need to be considered and reflected in the software design for opportunistic network applications. Together with the more obvious functionalities, for example, mutual presence awareness of nearby nodes, we formulate a number of adequate building blocks for opportunistic network applications. Described as services, they are integrated in our opportunistic network reference architecture (see Chapter 6). The modularization allows a concrete application design to combine and use the subset of services that are appropriate for its requirements.

Presence Awareness Service

This service provides the application with information about other nodes and users that are currently active and in communication range. Typical information includes a unique node ID and a timestamp about the last successful communication.

Message Exchange Service

A service that allows messages to be sent and received from nodes in communication range. This service implements the core *one-hop* communication paradigm as introduced earlier. This service does not guarantee a successful and errorless message delivery, since node mobility may always introduce unrecoverable link breaks during communication. Message delivery acknowledgements needs to be done at the application layer. The service does not make any assumptions about the message payload. The payload depends on the concrete application.

Information Filtering Service

Since in opportunistic network applications there is also the danger of SPAM, there should be a way to filter out information that might not be relevant to the user. This functionality is provided by an information filtering service. This service makes use of the information tagging and filters as described in Chapter 3. The authenticity of commercial information is secured with digital signature and provided by the security services (see below).

Information Distribution Service

The information distribution service offers three functional choices. A node can give information it receives straight away to other nodes in communication range. The user may also review a received piece of information and decide on a per item basis whether to share it with other nodes. Finally, the information may not be shared at all. This service is suitable for applications where an incentive scheme is not appropriate, i.e., if the application does not reward users' participation, this service allows a user to share or restrict his device resource.

Security Service

In order to support data or communication integrity and authenticity, the security service offers sign and encrypt operations on information. This may involve public-key cryptography operations, based on some PKI or other trusted sources. A service implementation needs to take the computational power of the target device into account. Some cryptographic operations or available algorithms are too complex or run too slow on mobile, battery powered devices.

Identity Management Service

The system design has to specify how a user appears in the system. Users can act anonymously, under a pseudonym or with assigned identities. The identity management service supports this design criterion. For anonymity, this service needs to take care of static data avoidance in all communication layers. Otherwise, a user might be tracked on a lower communication layer. This design issue is discussed in detail in Chapter 5.

User Notification Service

This service instantly notifies the user of incoming information that may require some sort of instant reaction. For example, a real-life conversation with a discussion partner can only happen while the partner is nearby (see *active collaboration* below). Implementing this service depends on device capabilities and notification urgency. A less urgent message may be flagged by the device blinking or vibrating in short for a couple of seconds, while

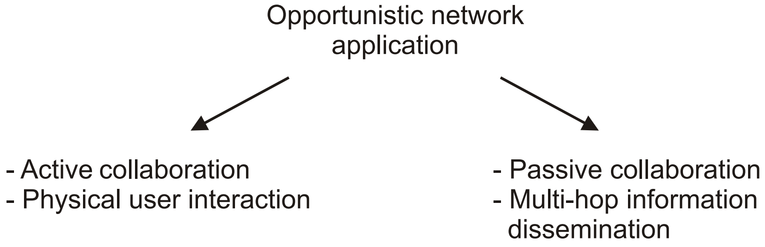

Opportunistic network application
- Active collaboration
- Physical user interaction
- Passive collaboration
- Multi-hop information dissemination

Figure 2.2: Design space of opportunistic network applications

urgent messages might cause the device to play back a high sound over and over again.

There are two more services that need to be custom-tailored for each application. One service implements our *incentive scheme* and the other service is a fixed node extension, in order to realize *proximity based services*. Examples of these services are described in Chapter 6 as part of the adPASS prototype description.

Table 2.3 summarizes the common services and their conceptual usage in the related work. The number of defined services is sufficient to cover the provided functionality of prior work. Notably, most related research work overlook security and identity management functionality. Both issues are crucial to preserving user privacy, the first important human aspect addressed within this work, and consequently increasing user acceptance in a concrete application.

On top of this, for adPASS, this thesis addresses incentive as the second important human aspect. Our incentive scheme is applicable to other domains as well (see Section 5.2.13). Finally, the opportunity to realize a *proximity based service* is addressed by few other works.

2.5 Design Space

The opportunistic network design space for applications is divided into two general areas as shown in Figure 2.2:

- **Active Collaboration** exploits the physical proximity of users. In addition to the exchange of digital information with users nearby, this allows the device to be used as a *link* to the user itself. Via non-intrusive user notification, for example a subtle device vibration, users are made aware of each other. This may lead to face-to-face collaboration, e.g., a conversation or a common goal pursuit.

	Presence Awareness	User Notification	Message Exchange	Data Filtering	Data Dissemination	Security	Identity Management	Incentive Scheme	Proximity based LBS
[Bea05]	✓	✓	✓	✓	✓				
[BBH02]			✓	✓	✓				✓
[DQA04]			✓	✓	✓				
[Eag05, EP06]	✓	✓	✓	✓					
[GFH05]			✓	✓	✓				
[GSX02]			✓	✓	✓				
[HW05]			✓	✓	✓				
[HFW99]	✓								
[Iwa98]	✓	✓		✓					
[KBTR02]			✓		✓				
[KLW03, KLW04]			✓		✓				
[LW02a, LW02b, LW05, Wal05]			✓	✓	✓				
[RSCH04]			✓	✓	✓				✓
[SHCD06]			✓		✓				
[SB03, SBB04b, SBB04a, SBB05]	✓	✓	✓	✓					
musicClouds	✓		✓	✓	✓				
adPASS			✓	✓	✓	✓	✓	✓	✓

✓= Usage

Table 2.3: Usage of opportunistic network building blocks

	Active Collaboration	Passive Collaboration
Presence Awareness	✓	✗
User Notification	✓	✗
Message Exchange	✓	✓
Data Filtering	✓	✓
Data Dissemination	(✗)	✓
Security	(✗)	(✓)
Identity Management	✗	(✓)
Incentive Scheme	✗	(✓)
Proximity Based Service	✗	(✓)

✓= yes, ✗= no

Table 2.4: Design space characteristics

Active collaboration has the advantage that the complete knowledge of a user does not need to be stored on the device itself. A short summary or some keywords are sufficient. Deeper knowledge about a topic may be exchanged by other means, after initial contact between users has been made by their devices.

- **Passive Collaboration** is a means to collect and pass any kind of information from and to other users in communication range. This happens without any user interaction.

  Passive collaboration leads to autonomous information dissemination. In other terms, it is a form of digital *word-of-mouth* communication.

  Since user devices act without user control and interference, an incentive scheme might be crucial for the application acceptance, due to the fact that users share private resources (memory, battery, CPU). Otherwise, a user might not be interested in taking part at all.

Each area demands a certain subset of building blocks to operate satisfactorily. This is depicted in Table 2.4. The entries in brackets indicate that the presence or absence of a functionality is not fixed. Overall, it is a good design choice to include or exclude a functionality as proposed in Table 2.4 when designing an opportunistic network application for active or passive collaboration respectively. But then there might be good reasons to deviate from this. For example, the presence of an incentive scheme is a good choice to attract users. On the other hand, if the application targets purely altruistic users, this might be omitted to attract the right kind of people. For example, an application that disseminates certain political ideas might live without an incentive scheme and is based simply on the user's political attitude.

Having defined the areas for opportunistic network applications, it allows us to classify related work into one domain or the other:

- **Active Collaboration**

  [Bea05], [Eag05, EP06], [GFH05], [HFW99], [Iwa98], [SHCD06], [SB03, SBB04b, SBB04a, SBB05]

- **Passive Collaboration**

  [BBH02], [DQA04], [GSX02], [HW05], [KBTR02], [KLW03, KLW04], [LW02a, LW02b, LW05, Wal05], [RSCH04]

2.6 Summary

This chapter discussed technical background and related work relevant to this thesis. We formulated a number of criteria crucial for opportunistic networks and reviewed work of others that expose similar ideas and concepts. We identified a number of building blocks for opportunistic networks whose functionality is sufficient to cover related work. Our opportunistic reference architecture presented in Chapter 6 integrates these building blocks.

In addition, our analysis helps to span the opportunistic network application design space. We distinguish between *active* and *passive* collaboration and group related work accordingly.

What is missing in state-of-the-art research though, is a coherent view that covers all aspects of opportunistic networks, in other words, to view opportunistic networks in its entirety. This is addressed in the following chapter.

Chapter 3

Opportunistic Network Concepts

In the last chapter, we have seen that opportunistic network research and related projects exploit ad-hoc short-range communication among mobile devices in different manners. In the passive collaboration domain, one can distinguish between two related but subtly different opportunistic network approaches: The use of user and thus device mobility for i) opportunistic network message routing/forwarding or for ii) opportunistic data dissemination. Recall, that the former assumes an end-to-end communication need between two or more communication partners but without a direct path between the endpoints, while the latter does not.

This chapter addresses opportunistic network concepts with a focus on data dissemination and user acceptability. For the data dissemination process, humans carry mobile devices around with them and the devices exchange data in a spontaneous manner, whenever they come close. Moreover, a human-centric view raises user acceptability issues in terms of privacy and incentives. We look into these issues in their entirety by presenting our definition of an opportunistic network and its components with *one-hop* communication as its core (Section 3.1). This leads us to an opportunistic system model as described in Section 3.2. After that, Section 3.3 discusses interaction patterns and communication semantics between the opportunistic network components. Then, a description on how information propagates through the network is given in Section 3.4. Basically, we distinguish between three information dissemination mechanisms, namely *one-hop information pass*, *time shifted information pass*, and *information move*. Section 3.6 outlines our approaches and concepts to increase user acceptability. These are presented in greater detail in Chapter 5. We summarize this chapter in Section 3.7.

3.1 Terms and System Components

A priori, opportunistic network applications do not make any assumptions about the participating users and the relation among the users. Thus, the opportunistic network model applies to a spontaneous network of humans that are, in the most general case, anonymous to each other. The model exploits the physical proximity of nodes. In the real world, physical proximity of humans allows for a conversation and, at least, indicates a situational affinity between humans. This affinity is mapped to the user's device and used for information exchange. This idea is captured in our *opportunistic network* definition below.

Definition 1 (Opportunistic Network) An *opportunistic network* is a network of wirelessly connected nodes. Communication range between two connected nodes is not further than walking distance. Nodes are connected only temporarily and the network topology may change due to node mobility or node activation and node deactivation respectively. The network provides at least the following functionalities:

- **Node discovery**.

  A network node is able to discover other network nodes in direct communication range.

- **One-hop message exchange**.

  A node is able to send and receive arbitrary data to or from any other node in direct communication range. □

In other words, in an opportunistic network, there is the *opportunity* for nodes to recognize other nodes in physical proximity and to 'talk' to them.

Definition 2 (Opportunistic Network Node) An *opportunistic network node* is a device with short-range wireless communication capabilities. The device runs an opportunistic network application that uses a data sharing protocol for data dissemination. The data sharing protocol is based on i) node discovery and ii) one-hop message exchange as stated above. □

Definition 3 (Mobile Node) A *mobile node* within an opportunistic network consists of a user carrying a mobile device that acts as an opportunistic network node.[1] □

Definition 4 (Information Sprinkler) An *Information Sprinkler* (IS) is a fixed opportunistic network node within the network. It is a device placed at a dedicated location, thus it is not mobile and not under direct user control. The Information Sprinkler uses the same data sharing protocol as other opportunistic network nodes. □

It can operate in a *sprinkler* mode, meaning information is only dispersed, or in a *sink* mode, meaning information is only collected, or in both modes together. An Information Sprinkler may also be connected to a *sprinkler backbone* network. The backbone network may be a wired network that connects a set of Information Sprinklers and synchronizes their operation. For example, data that is collected at one Information Sprinkler becomes available at all other sprinklers soon after.

[1] If the wording is clear from the context, we use the term *mobile node* and *node* interchangeably.

This thesis also uses the term *Information Sprinkler* if the device runs only in *sink* mode or in a combination of both.

The next two definitions introduce two core node components that are essential for our data dissemination protocol.

Definition 5 (Node Profile) A *node profile* is a data structure stored on an opportunistic network node. Using the profile, a node specifies what information it is interested in and what information it wants to share with other nodes. For this purpose, the node profile splits into two sub-components, the so-called *information lists* (iLists):

- *iHave-list* (information have list):

  The iHave-list holds all the information the node wants to contribute to other nodes. A single entry on the iHave-list is called *information item*.

- *iWish-list* (information wish list):

  In the iWish-list, the node specifies what kind of information it is interested in. A single entry on the iWish-list is called *information wish* or *wish*. □

Definition 6 (Neighborhood) A *neighborhood* is a node's software component that keeps track of other active nodes in the vicinity, i.e., successfully discovered nodes. □

The process by which information items are distributed among nodes within an opportunistic network is called *data dissemination*. Rules about distribution are specified by a data sharing protocol that makes use of the node profile. The data sharing protocol is based on two steps: i) node discovery and ii) exchange of information lists. This is elaborated upon in Section 3.3.

3.2 Node Architecture, System Model and Proximity Based Services

The definitions from the last section, together with the identified opportunistic network services from the last chapter, lead us to an architecture for opportunistic network nodes and a corresponding system model, which we will describe now.

3.2.1 The iClouds Architecture

In order to develop and investigate opportunistic network concepts, the *iClouds* project [Hei07] was set up at the Telecooperation Group (Computer Science Department, Darmstadt University of Technology). iClouds is an abbreviation for *information clouds*. Imagine a pedestrian walking around a city center and encountering other pedestrians. Each person he passes could be a potential information

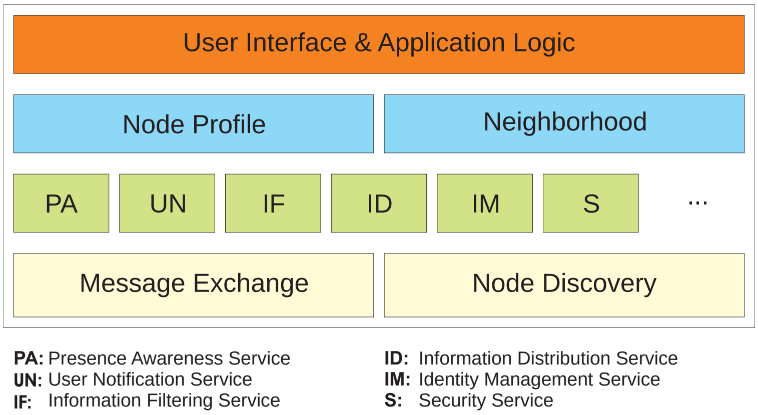

User Interface & Application Logic
Node Profile
Neighborhood
PA
UN
IF
ID
IM
S
...
Message Exchange
Node Discovery
PA: Presence Awareness Service
UN: User Notification Service
IF: Information Filtering Service
ID: Information Distribution Service
IM: Identity Management Service
S: Security Service

Figure 3.1: Node architecture

bearer. Thus, our user wanders through an imaginary cloud of information or information cloud. This metaphor grasps the fundamental idea of collaboration in opportunistic networks. The effortless sharing of information by passing messages in a spontaneous manner.

The goal of the iClouds project was to investigate methods for sharing information among a group of users, based on individual user contribution. The opportunistic network dissemination protocol is such a method.

As a part of iClouds, an architecture for opportunistic network nodes was developed, shown in Figure 3.1. The architecture reflects the common building blocks from Chapter 2 (see Section 2.4) and the definitions from the last section.

We distinguish between four different layers. The bottom layer handles simple communication issues, i.e., adjacent node discovery and one-hop message exchange between nodes in communication range. For example, the neighborhood data structure on the third layer makes use of the node discovery mechanism.

The common services are located on the second layer. Each service can use functionalities provided by other services or by the bottom communication layer. Note that the service layer is extensible for new services that might be needed by future applications.

The node profile and neighborhood data structure, being present in all opportunistic network applications, resides on the third layer.

An application's specific logic and user interface reside on the topmost layer. To fulfill its purpose, this layer has access to all layers below.

3.2.2 System Model

The opportunistic network nodes define our opportunistic network system model. An example of this model is depicted in Figure 3.2. The figure shows three Infor-

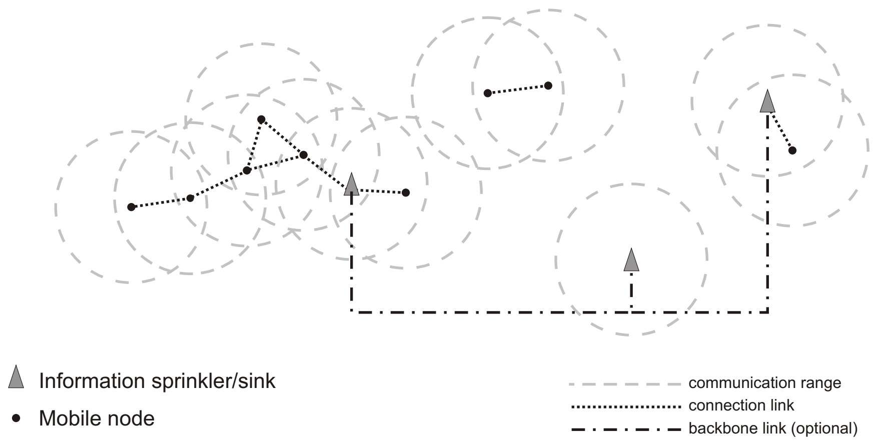
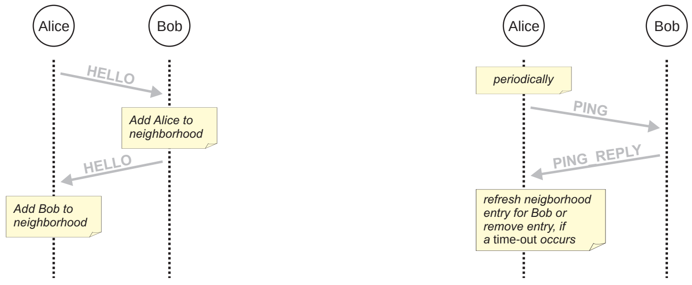


Figure 3.2: System model for information dissemination

mation Sprinklers and the optional sprinkler backbone. A connection link between nodes is indicated by a black dashed line. It shows several mobile nodes with their communication ranges (dotted sphere). Note that in practice, the communication range of a node is not an ideal sphere, due to communication signal interferences with the surroundings. For example, in city settings, buildings will reduce communication range, whereas in a park with direct line of sight, communication range will not be harmed (see Chapter 6.2).

3.2.3 Proximity Based Services using Information Sprinklers

As already stated in Chapter 2, with our opportunistic network system model, a simpler form of a Location Based Services is possible. We call it *proximity based services*. It exploits the physical proximity of a user to an Information Sprinkler. Since an Information Sprinkler is set up at a dedicated place, it can store its geographical location and provides an information service that is useful for that location. By passing an Information Sprinkler, a user's device learns its current geographical location and is provided with information that might be useful to the user. For example, an Information Sprinkler located at the entrance of a shopping mall might disseminate the latest advertisements belonging to the various shops at the mall. The adPASS system (Section 6.1.1) uses an Information Sprinkler for this purpose.


Alice
Bob
HELLO
Add Alice to neighborhood
HELLO
Add Bob to neighborhood
Alice
Bob
periodically
PING
PING_REPLY
refresh neigborhood entry for Bob or remove entry, if a time-out occurs

(a) Alice and Bob discover each other

(b) Check node liveliness

Figure 3.3: Node discovery and liveliness check

3.3 Communication Semantics

As stated in Section 3.1, an opportunistic network builds on node discovery and one-hop message exchange. Leaving technical realization details aside, this section describes the principal communication interaction and methods for node discovery and data sharing.

3.3.1 Node Discovery

The bootstrapping phase in opportunistic network communication is node discovery. A node needs to identify other nodes in its vicinity in order to start collaboration. This may be a form of user notification that leads into an active collaboration or the execution of a data sharing protocol for the purely passive collaboration of devices only.

The discovery process makes use of the neighborhood software component. Periodically, each node announces its presence by broadcasting a HELLO message, illustrated in Figure 3.3(a). This message includes a unique node ID. Whenever a node receives a HELLO message, it updates the neighborhood component. Suppose node Bob receives a HELLO message from node Alice at time t_{seen}. If the neighborhood component of Bob does not have an entry for Alice, a new entry is added. The new entry includes the unique ID of the new node and the timestamp t_{seen}.

For an existing entry, the last-seen timestamp is replaced by t_{seen}. Node Alice will act in the same manner, if it receives a HELLO message from node Bob. The t_{seen} timestamp is checked periodically by the node. A value that is outdated according to a node's settings indicates that the entry may be deleted, because the other node has either moved out of communication range or has been turned off. Before the entry is removed, a node checks this assumption and sends a PING message to the other node. If the other node answers within a defined time frame, the entry is refreshed. Otherwise the entry is removed (see Fiz‘gure 3.3(b)).

3.3.2 Data Sharing

After two nodes have successfully discovered each other, both nodes need to find out if it is beneficial for them to communicate further, i.e., to make the device owners aware of each other or to simply share information. For both tasks, the node profile is used. Recall that the node profile consists of two data structures. The iWish-list holds information a user is interested in and the iHave-list stores information a user is willing to share.

Our proposed data sharing protocol is based on exchanging information lists between connected nodes. Items on the iWish-lists are matched against items on the iHave-lists. On a match, information items move from one iHave-list to the other.

Again, consider two nodes, Alice and Bob, who meet on the street. When the nodes discover each other, they might exchange their iHave-lists and match them locally against their iWish-lists. If an item on Bob's iHave-list matches an item on Alice's iWish-list, her device will transfer that item onto her iHave-list.

For two nodes that are in communication range, there are two communication methods for transferring the iLists. Nodes can either *pull* the iLists from other nodes or they can *push* their own iLists to nodes they encounter. In addition, either of these two operations is applicable to both lists, which gives us four distinct possibilities of communication. We summarize these possibilities, along with their real world equivalents, in Table 3.1.

	pull (from Bob)	push (to Bob)
iHave-List	Standard search	Advertise
iWish-List	Active service inquiry	Active search

Table 3.1: Information flow semantics (from Alice's point of view)

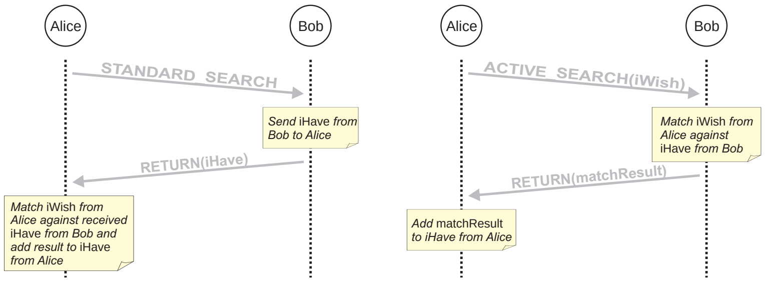
In each of the four cases shown in Table 3.1, the matching operation is always performed on the peer who receives the list (Alice's in pull and Bob's in push). Each of the four possible combinations corresponds to some interaction in the real world:

- *Standard search.* Alice pulls iHave-list from Bob.

  This is the most natural communication pattern. Alice asks for the information stored on Bob's device and performs a match against her information needs (specified in her iWish-list) on her device. We can also see the user as just passively "browsing" what is available (Figure 3.4(a)).

- *Advertise.* Alice pushes her iHave-list to Bob.

  This is a more direct approach. Alice gives her information items straight to Bob and it is up to Bob to match this against the things he is interested in. As an example, consider an Information Sprinkler mounted on shopping mall


Alice
Bob
STANDARD SEARCH
Send iHave from Bob to Alice
RETURN(iHave)
Match iWish from Alice against received iHave from Bob and add result to iHave from Alice
Alice
Bob
ACTIVE SEARCH(iWish)
Match iWish from Alice against iHave from Bob
RETURN(matchResult)
Add matchResult to iHave from Alice

(a) Standard search

(b) Active search

Figure 3.4: Node behavior for standard and active search

doorways transmitting advertisements to customer devices when they enter the building.

- *Active service inquiry.* Alice pulls iWish-list from Bob.

  This is similar to shopping clerks. They learn at a very early stage, what their customers are interested in. An example of this query could be: "*Can I help you? Please show me what you are looking for.*"

- *Active search.* Alice pushes her iWish-list to Bob.

  With active search, we model the natural "*I'm looking for X. Can you help me?*" This is similar to the standard search mechanism, except that the user is actively searching for a particular item, whereas in the standard search the user is more passive (Figure 3.4(b)).

Figure 3.4(b) depicts the communication interaction for the *active search* case. First, Alice sends her iWish-list to Bob. Bob compares the expressed information wishes against the information he offers, i.e., the information stored on his iHave-list. Successful matches are sent back to Alice who in turn adds the newly learned information to her iHave-list.

Exchanging iLists as described above has direct implications on user privacy. A mechanism for preserving privacy is discussed in Section 3.6.1. Inherently, it depends on the concrete application whether user privacy is in danger by participating in the network. For instance, an application that shares tourist information may be of lower privacy concerns to a user than an application that shares digital advertisements. With the first application, for example, an observer learns that there is someone in the vicinity that offers information about the opening hours of the local museum. The second, in contrast, tells more about the user in the vicinity. If the application offers an advertisement about a DVD player, this implies that the user is interested in DVD players and therefore DVDs as well. Further, if an observer is able to identify this user, he might sell the name and postal address to a DVD seller, who might in turn send unwanted commercial mail. The information, *'Alice likes DVDs'* might have a higher value to an observer than *'Alice knows the opening hours of the local museum.'* Anyway, in both cases, it should be up to the user to reveal her identity. The mechanisms discussed on page 50 address this issue.

From the privacy viewpoint, both methods for transferring iLists are considered equal. In order to exchange information, Alice has to give away her iHave-list or her iWish-list. From the iWish-list, an observer learns directly what information Alice is interested in. However, from the entries on the iHave-list, an observer can also derive what information Alice is interested in.

3.4 Data Dissemination Mechanism

This section looks closer at the data dissemination mechanisms that derive from the system model and communication semantics. As stated in the first chapter, the proposed scheme deliberately sets *multi-hop* end-to-end communication between nodes aside. Therefore, no routing mechanism needs to be supported. Data dissemination relies solely on *one-hop* communication and uses a node's profile to carry out its task. This happens without user interaction. The dissemination process is based on consecutive one-hop communication events between directly connected nodes. These nodes, after they have detected a match in their node profiles, exchange data with each other. The physical and independent movement of nodes is utilized to distribute the data. Conceptually, we distinguish between three mechanisms, which we will describe now.

3.4.1 Information Pass

The basic mechanism is called *information pass* and is illustrated in Figure 3.5(b). As the name suggests, some information is passed from one node to another. For this, the following conditions must be fulfilled.

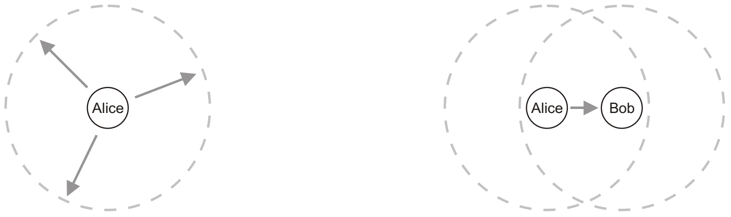
- Nodes must be within communication range (Figure 3.5(a)).
- Node Alice offers information that node Bob is interested in.

For any two nodes that are within communication range, we can conclude that the nodes are in close proximity at the same time. Basically, they are at the same place at the same time. According to the information exchange protocol used, two nodes match their profiles. If an entry on the iHave-list of Alice matches an entry on the iWish-list of Bob, this entry (information) is passed from one node to the other.

In Figure 3.5(b) Alice is in communication range to Bob and passes information to Bob (indicated by the arrow). This may also happen in the other direction simultaneously.

3.4.2 Time-shifted Information Pass

A variation of *information pass* is called *time-shifted information pass*. This mechanism uses an Information Sprinkler, enabling users to share information who are at



(a) Communication range

(b) Information pass

Figure 3.5: One-hop information pass

the same place but at a different time. As an example, consider a user Alice who goes to a local coffee bar at 10 every morning. User Bob visits the same place each afternoon. Alice and Bob will never meet and thus come into communication range while visiting that coffee bar. In this situation, the installation of an Information Sprinkler helps. The sprinkler is set up in the bar and collects all information of users visiting the bar. This allows Alice to leave her information at the sprinkler in the morning and Bob to learn about this information from Alice in the afternoon. Figure 3.6 depicts this mechanism.

In order to reduce communication costs and storage capacity at the Information Sprinkler, the mechanism might be optimized in the following way. Bob leaves his information wishes at the sprinkler. Later Alice asks the sprinkler for new wishes and matches these against her information. Then, the successful matches are passed from Alice to the sprinkler. Assuming that Bob visits the coffee bar in the afternoon and his information wishes have not changed since the last visit, the sprinkler can now pass the information from Alice to Bob. Therefore, storage for information that Bob is not interested in is not wasted at the sprinkler and also not transmitted to it in vain.

3.4.3 Information Move

Information move is based on *information pass* and user mobility as illustrated in Figure 3.7. At first, Alice and Bob are within communication range and Alice passes information to Bob in which he is interested. Then, Bob and Alice part and later on Bob comes into communication range of Claire. Assuming that Claire has an interest in the same information, Bob will pass the previously collected information on to Claire. Therefore, the information is disseminated among interested nodes. For the information dissemination effectiveness, see the simulation results in Chapter 6.

3.5 Information Tagging and Information Filters

As described in Section 5, each node stores a node profile for the data dissemination task. Recall that the iHave-list stores information a node already knows and the

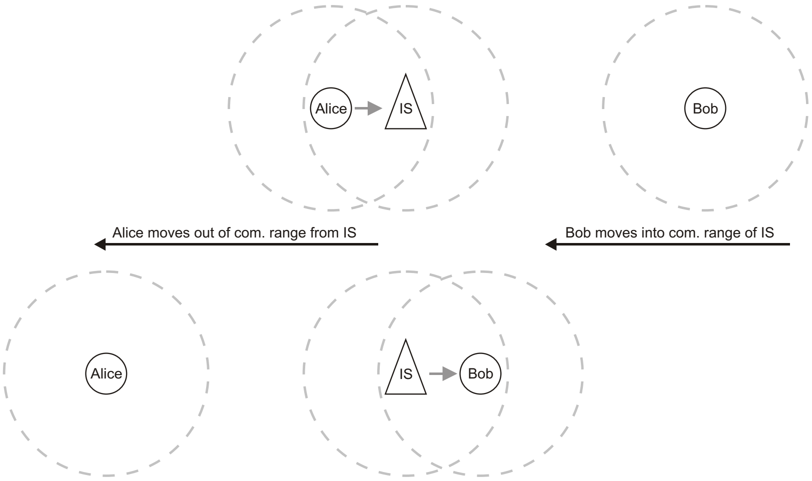

Alice
IS
Bob
Alice moves out of com. range from IS
Bob moves into com. range of IS
Alice
IS
Bob

Figure 3.6: Time-shifted information pass

iWish-list expresses a node's interest in information. Therefore, it must be possible to *match* a given information item against an expressed interest. This is achieved by

1. *Tagging* the information item with *attributes* from a given attribute set.
2. Express information interest as a set of logical filter expressions. Each term represents a filter that yields `false` or `true` when applied to an attribute.
3. Each information item is examined in turn. If a logical filter expression exists that matches the considered information item, i.e., the filter expression is evaluated as `true`, the information item is passed from one node to another.

The following illustrates information tagging and information filters by means of an example. This approach is formalized in Chapter 4.

Example Consider an application that disseminates digital music in form of mp3 files among nodes. The *musicClouds* application (see Section 6.1.2) has implemented this functionality. Tagging a file is done by the ID3-tag standard. An excerpt for a pop song by the artist *Madonna* is shown in Table 3.2. The table also lists corresponding attribute data types and suitable operators for the data type. This information is needed to evaluate the filter terms.

The following filter, given in a pseudo formal notation, expresses interest in all songs by Madonna that were released after the year 1999. Therefore, it would match the Madonna song from above.

`(` Artist `equals` *Madonna* `)` `AND` `(` Year > *1999* `)`

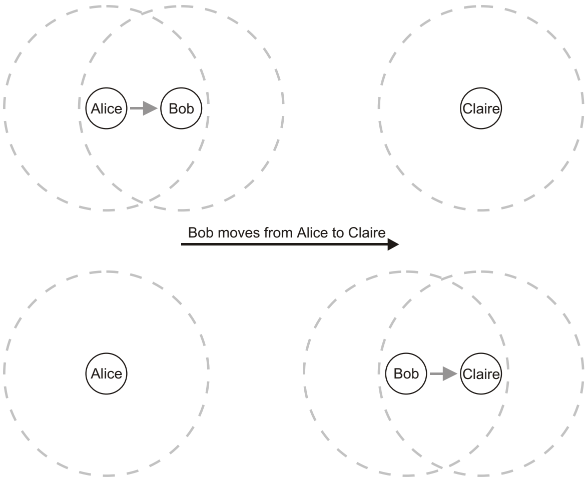


Figure 3.7: Information move

Another example expresses interest in the pop and disco genre (encoded by the ID3v1 constants 13 and 4).

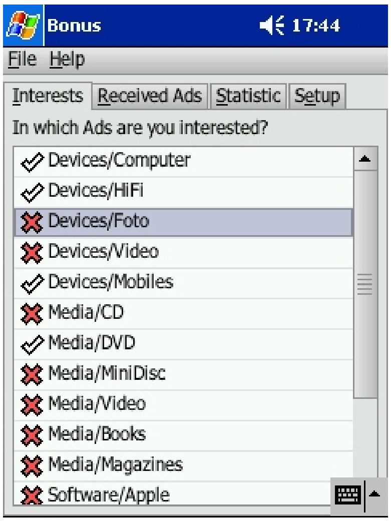
(Genre = *13*) OR (Genre = *4*)

The formal model, presented in Chapter 4, allows the usage of Boolean operators AND, OR, and NOT within logical filter expressions. Please note that the readability and expressiveness of an expression depends on the defined data types and corresponding operators. For example, if an application allows for data type `set` and the standard set operators ∈, ⊂, ∉, the second filter could be expressed in following way:

Genre $\in \{4, 13\}$

Attribute	Value	Data type[2]	Operator
Song title	*Gone*	`STRING`	`equals, startsWith, endsWith`
Artist	*Madonna*	`STRING`	`equals, startsWith, endsWith`
Album	*Music*	`STRING`	`equals, startsWith, endsWith`
Year	*2000*	`DATE`	`=,<,>`
Comment	*my favorite*	`STRING`	`equals, startsWith, endsWith`
Genre	*13*[3]	`BYTE`	`=`

Table 3.2: ID3v1 tag example



In which Ads are you interested?

- ✓ Devices/Computer
- ✓ Devices/HiFi
- ✗ Devices/Foto
- ✗ Devices/Video
- ✓ Devices/Mobiles
- ✗ Media/CD
- ✓ Media/DVD
- ✗ Media/MiniDisc
- ✗ Media/Video
- ✗ Media/Books
- ✗ Media/Magazines
- ✗ Software/Apple

Figure 3.8: adPASS screenshot with category selection tab

3.5.1 Information Categories

The opportunistic network model and its corresponding data dissemination mechanism is applicable to a variety of applications. After the purpose of a certain application has been defined, the application designer has to specify a set of suitable tag attributes. For each tag attribute, the appropriate data type has to be chosen. As described in the last section, this has implications on filter expressiveness.

The implementation task raises another important issue for the programmer. How should the information be presented to the user and how should a user type in his filter expressions. Obviously, an application will increase its usability, if it is easy to browse collected data objects. An established and well understood approach is to define a *category system*. This allows information to be presented, sorted, or selected according to that system. You can find examples for this approach in web catalogs like Yahoo [Yah05] or the open directory project [Com07], online shops, for example Amazon [Ama95], or online auctions like eBay [eBa95].

Within the concept of tagging, a category is just a special attribute. Regarding the musicClouds application, the *genre* tag is a suitable candidate for a category system. This approach is also used at the musicmoz site [Var05] under the name *Music by Style*.

Similarly, you can find easy and fast user navigation in online shops by organizing products according to a class of product categories. This natural approach was also chosen in the adPASS prototype implementation illustrated in Figure 3.8.

.

[2] These are not the data types as specified by the ID3v1 standard. `STRING` is actually `30 CHARACTER` and the attribute `Year` is also `30 CHARACTER` and not a `DATE`. This is changed for a better illustration of suitable operators.

[3] According to the ID3v1 specification, a value 13 defines *pop music*.

3.6 User Acceptability

As was said in the introductory chapter, user acceptability may be increased by addressing the issue of *privacy* and *incentives*. We will look into both issues in turn. Each will be discussed in terms of why privacy and incentives are important. The results of the analysis will be assisted by our opportunistic network architecture described in Chapter 6. The adPASS prototype serves as a proof of concept implementation (see Section 6.1.1).

3.6.1 Privacy Preservation

Mobile nodes in an opportunistic network are carried by humans (see Definition on page 36). Given that communication happens in a user's physical proximity and that the user's device will pass information or information wishes without notice, this may conflict with user privacy needs.

Privacy is the ability of a user to stop information about himself from becoming known to other users. In the realm of opportunistic networks, it should be possible for a user to express an information wish or offer an information item to others without creating the possibility that this action can be linked back to himself, especially for applications that aim at *passive* collaboration. This requirement might be different for *active* collaboration applications. An application that aims to bring people together needs one way to identify users and therefore, needs to breach privacy, while pure information dissemination applications may have a higher user acceptance if privacy is protected. For example, in adPASS, users express interest in certain kinds of products. It makes a significant difference whether an observer learns that *"A user in my vicinity is interested in high end DVD player"* or *"A user named* ***Alice*** *in my vicinity is interested in high end DVD player"*. The second may lead towards learning more about Alice, for example her postal address. This could be used to estimate her living standards and so on. Whether this information is purely used to offer Alice additional high end accessories for her DVD player or it is used for other more dubious actions is at this point irrelevant. Privacy preservation should be an option for opportunistic network applications.

In order to breach a user's privacy, gathered data has to be linked to the human being in the real world. This involves identifying the person. There are three distinguishable degrees to classify user identifiability (see Figure 3.9):

- **Identity**

  A user that communicates with others and reveals any piece of information that can be used to clearly identify him is said to work under his *identity*. Examples are the full name of a user (if not too common) or his social security number.

- **Pseudonymity**

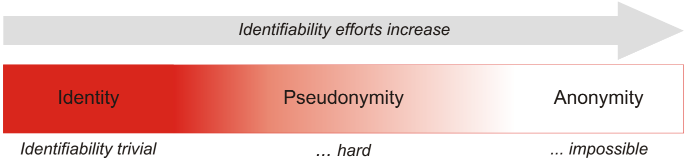


Figure 3.9: Degrees of user identifiability

This is the ability to prove a consistent identity without revealing a user's real identity, instead using a *pseudonym*. This is very common on the Internet, for example in chat rooms or with electronic mail. Users are free to choose a nickname as a pseudonym and identify themselves with that.

Whether a pseudonym can be linked to the real identity of a user depends on a variety of factors. For example, while it may be impossible for you to identify other members in a chat room, this may be trivial for the Internet Service Providers.

The harder it is to reveal the pseudonym of a user, the closer we are to the state of not being identifiable at all, thus acting anonymously.

- **Anonymity**

  Anonymity is the ability to remain unidentifiable within a set of users. A user acts *anonymously* if it is impossible to reveal his identity.

As stated above, different applications involve different degrees of user identifiability. In addition, some applications may ask for anonymity and provableness simultaneously. The proposed incentive scheme (see below) depicts such an example. In order to achieve these goals, this work proposes two combined technical solutions that correspond with opportunistic network characteristics.

- A network node waits for a minimal amount of users in its proximity before taking part in information sharing (goal: anonymity).

  If there is a minimal set of nodes active at the same place and time, it is harder for an attacker to deduce the source of a certain information item or information wish respectively.

- A node changes its network identifier frequently (goal: anonymity).

  Since opportunistic networks are based on *one-hop* communication and do not use routing or any other multi-hop message exchange, it is feasible for a node to generate its network identifier by itself and to change this identifier periodically. Thus, it is harder for an attacker to map communication behavior to one particular node over the course of time. This approach also defeats user

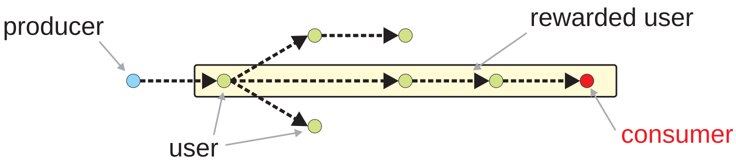


Figure 3.10: Incentive scheme – basic idea

movement tracking, that would otherwise be possible with a unique device ID once an attacker has revealed a user's name.

- A node uses public keys as aliases (goal: provableness).

  A node generates a set of key-pairs (RSA keys for example). Each message exchange is tagged with a public key as an alias and signed with the corresponding private key. A signing operation in which the private key is used may be carried out later to prove the legitimate ownership of the public key.

3.6.2 User Incentives

Since the opportunistic network model exploits the private resources of users, mainly battery power and device memory, the question arises why a user should take part in the system at all.

An obvious benefit for the user is the potential fulfillment of his information needs. The device collects only information the user is interested in (see *complex filter* Definition 11 in Chapter 4). In return, this information is shared with others. Peer-to-Peer file sharing applications work in a similar manner.

In addition, the opportunistic network model can be extended with an incentive scheme to stimulate and thus increase user participation. In short, the proposed incentive scheme allows users to gain some kind of benefit by passing information on to other users. The incentive scheme is formalized in Section 5.2. Here we will present the idea, participants and interaction pattern.

For the time being, we only sketch the incentive scheme, the underlying idea and its usage within the adPASS-prototype. A more general definition on and usage of the incentive scheme is given in Chapter 5. There you will also find an usage example outside the domain of opportunistic networks (see page 93).

Basic Idea: The incentive scheme rewards users who partly help to carry an information item from a producer to a consumer. Here, 'carry an information item' includes information pass, time-shifted information pass and information move (see Section 3.4).

Figure 3.10 illustrates this. A producer, for example an Information Sprinkler that disseminates digital advertisements inside a shopping mall, passes information to nodes in communication range. Recall that the information item is passed on only if an entry on the iWish-list matches the information item. In the example, the information item is passed on from node to node until it reaches a consumer that uses the information item for his benefit. A received advertisement could stimulate the consumer to buy the advertised product at the producer's shop. Thus, all nodes framed in the yellow box should be rewarded by the producer or the consumer, since they helped in bringing them together. The set of nodes framed in the box is called *bearer chain*.

Roles: The incentive scheme distinguishes different roles of opportunistic network participants.

- An *Information Producer* ('producer' for short) is a node that creates information items and initiates their dissemination.
- An *Information Bearer* ('bearer' for short) receives an information item from an Information Producer or another Information Bearer and passes it on to other Information Bearers or Information Consumers. An Information Bearer is only interested in gaining a reward for transporting information.
- An *Information Consumer* ('consumer' for short) receives information from an Information Producer or an Information Bearer. He takes some action on it that is beneficial for himself and the Information Producer. This leads towards dispensing a reward to all Information Bearers that took part in carrying information down the 'Information Producer to Information Consumer' path.

  A node can act both as an Information Bearer and an Information Consumer. See adPASS (Section 6.1.1) as an example.

Mediator: Since the incentive scheme is based on the opportunistic network model and nodes act and move autonomously, there must be a way to issue a reward to bearers. A *Mediator* keeps track of the users' rewards. It is similar to a central database where the producer, the bearer, and the consumer have access to (for example via the Internet). Therefore, it guarantees accessibility to the inherently transient mobile nodes in the network. In addition, the Mediator helps to keep nodes anonymous, if they so wish. Again, see adPASS 6.1.1 for a concrete usage of a Mediator.

At this point, the incentive scheme conflicts with the privacy preservation of a user. Consider an arbitrary information item i. A user Alice who wants to claim a reward for being part of the bearer chain, thereby helping that i finds its way from the producer to the consumer, has to prove her participation in the bearer chain. This implies that Alice was interested in i and has shown this interest with a matching entry on her iWish-list. Therefore, the producer and the consumer might be able to learn, that a user Alice was interested in *i*.

To recapitulate, Alice should be able to prove her legitimate reward claim without revealing her identity. The proposed solution in this work makes use of public-key cryptography operations in the following way:

- A node generates a key pair on the device as a user pseudonym ($\approx$ public key) for each bearer chain.
- The public key is passed with the information *i* to flag the nodes participation.
- The private key is used to sign the bearer chain in order to prevent fraud and prove participation in the bearer chain.
- The private key is used to prove the legitimate ownership of the corresponding public key and thus used to claim rewards associated with the public key.

A detailed elaboration on the usage of public key cryptography as part of the incentive scheme is given in Chapter 5.2.

3.7 Summary

This chapter presented our core concepts for opportunistic networks. Having introduced system components and outlined the protocols and mechanisms on which data dissemination is based, the next two chapters will formalize the concepts. In Chapter 4, a formal model for information modeling, including generic matching algorithms is given. Later, Chapter 5 describes a mechanism to preserve user privacy within opportunistic networks, followed by the formalization of a generic incentive scheme.

Chapter 4

Formal Information Model

The data dissemination task in opportunistic networks requires matching user profiles by comparing entries on a user's iWish-list with entries on another user's iHave-list. For this purpose, information entities, called information items in this thesis, are tagged with metadata. Furthermore, filter expressions are applied to the metadata. Information tagging and information filters were briefly introduced in the last chapter.

In this chapter, the profile matching is formalized. In order to support a variety of applications, our model is *generic*, for example, it does not restrict itself to a certain technology, a certain data structure or programming language. In addition, the *expressiveness* of the model is sufficient to cover known similar applications, as described in Section 2.3.

This chapter presents basic definitions of our model. In order to get a better understanding, each definition is followed by an illustrative example. The examples are based on the musicClouds application (see Section 6.1.2).

The core concept of data dissemination is reflected in Definition 9 (Information Model), Definition 11 (Complex Filter) and Definition 12 (User Profile).

This chapter includes a simple top down illustration of how to use the model within an application development process (see Section 4.3 and a number of Java source code excerpts from the musicClouds application that serve as a model implementation example (Section 4.4).

4.1 Basic Definitions

For the model to be solid and sound, definitions for *datatype* and *value set* are given first.

Definition 7 (Datatype) A datatype D is a pair $D = (\mathbf{S}, \Omega)$, where $\mathbf{S}$ is a non-empty set of elements (values), `undef` $\notin \mathbf{S}$, and $\Omega = \{\omega_0, \omega_1, \ldots\} \subseteq \{0, 1\}^{\mathbf{S}\times\mathbf{S}}$ is a non-empty set of binary operators which can be applied to elements out of $\mathbf{S}$ and yield 0 (false) or 1 (true), formally $\omega_i : \mathbf{S} \times \mathbf{S} \to \{0, 1\}, \omega_i \in \Omega$.

Furthermore, there is at least one $\omega_j \in \Omega$ with $\omega_j(x, x) \mapsto 1$ for all $x \in \mathbf{S}$. We call ω_j equality operator and label it ω^{EQ}. □

We write $\mathbb{D}$ for a set of data types. Moreover we use the two restriction mappings operator () and domain () as follows:

$$\begin{aligned} \text{operator} &: \mathbb{D} \to \{\Omega \mid (\mathbf{S}, \Omega) \in \mathbb{D}\}, (\mathbf{S}, \Omega) \mapsto \Omega \\ \text{domain} &: \mathbb{D} \to \{\mathbf{S} \mid (\mathbf{S}, \Omega) \in \mathbb{D}\}, (\mathbf{S}, \Omega) \mapsto \mathbf{S} \end{aligned}$$

Example 1 (Data Type Set) The musicClouds application uses the following data types

$(\texttt{String}, \{==\})$	for the attributes: *Title*, *Artist*, *Album*, *Comment*
$(\texttt{int}, \{==,\ !=\})$	for the *Genre* attribute
$(\texttt{Date}, \{==,\ !=,\ <,\ >\})$	for the *Year* attribute

In all these data types, == defines the equality operator ω^{EQ}. The set $\mathbb{D}$ of data types is

$$\mathbb{D} = \{\ (\texttt{String}, \{==\}),\ (\texttt{int}, \{==,\ !=\}),\ (\texttt{Date}, \{==,\ !=,\ <,\ >\})\ \}$$

Applying the operator () and domain () mapping to $(\texttt{int}, \{==,\ !=\}) \in \mathbb{D}$ yields

$$\begin{aligned} \text{operator}\,(\,(\texttt{int}, \{==,\ !=\})\,) &\mapsto \{==,\ !=\} \\ \text{domain}\,(\,(\texttt{int}, \{==,\ !=\})\,) &\mapsto \texttt{int} \end{aligned}$$

□

Definition 8 (Value Set) The set of values, determined by $\mathbb{D}$ and used by the value () mapping (see Definition 9), is defined by

$$\mathbb{V}_{\mathbb{D}} = \bigcup_{(\mathbf{S}, \Omega) \in \mathbb{D}} \text{domain}((\mathbf{S}, \Omega)) \cup \{\texttt{undef}\}$$

□

Example 2 (Value Set) For the musicClouds application we have

$$\mathbb{V}_{\mathbb{D}} = \texttt{String} \cup \texttt{Date} \cup \texttt{int} \cup \{\texttt{undef}\}$$

□

At this point, a data type $\mathbb{D}$ and value set $\mathbb{V}$ serve to define an *information model*. The information model reflects the idea of tagging information with metadata in the form of (name, value)-pairs. Filters will use these pairs for matching information (see Definition 10 and 11).

Definition 9 (Information Model) An information model $\mathfrak{I}$ is a tuple

$$(\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$$

where

$\mathbb{I}$ is a set of information items,

$\mathbb{A}$ is a set of attributes,

$\mathbb{D}$ is a set of data types,

$\text{type} : \mathbb{A} \to \mathbb{D}$ is a mapping that assigns a data type to an attribute,

$\text{value} : \mathbb{I} \times \mathbb{A} \to \mathbb{V}_{\mathbb{D}}$ is a mapping that assigns a value to an attribute of an information item. □

An information item from the information item set $\mathbb{I}$ is labeled π, an attribute from the attribute set $\mathbb{A}$ is labeled α.

The information model definition formalizes information tagging. The information item set $\mathbb{I}$ specifies what kind of information an application supports. The set of attributes $\mathbb{A}$ defines suitable attributes for tagging an information item sufficiently. Using a set of data types $\mathbb{D}$, the type ()-mapping specifies an appropriate data type for each attribute. A data type may be a built-in data type of a chosen programming language or, if more complex, a custom data type implementation. An attribute of an information item is assigned a concrete value using the value ()-mapping. By convention, unknown attribute values are mapped to `undef`.

In summary, Definition 9 allows each information item to be tagged with a set of (name, value)-pairs. This serves as an anchor for the matching task, as we will see later.

Sometimes it makes sense to map an attribute onto several values. This might be modeled using a power set. Let $\mathbf{S} := \{s_1, s_2, s_3\}$ be an arbitrary set. A data type $D_0 = (\mathcal{P}(\mathbf{S}), \Omega)$ with values out of a power set allows assigning several values out of $\mathbf{S}$ to a certain attribute. For example, let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model with $\alpha_0 \in \mathbb{A}$, $D_0 \in \mathbb{D}$, and $\pi_0 \in \mathbb{I}$:

$$\text{type}(\alpha_0) \mapsto D_0 \tag{i}$$

$$\text{value}(\pi_0, \alpha_0) \mapsto \{s_1, s_3\} \text{ with } \pi_0 \in \mathbb{I}, \alpha_0 \in \mathbb{A} \tag{ii}$$

In general, (ii) is interpreted as an *and*-semantic, i.e., s_1 *and* s_3 is true for α_0. See the category attribute proposal on page 61 for an application.

The next example illustrates an information model. Again, it is based on the musicClouds application for music file dissemination.

Example 3 (Information Model) A snapshot of an information item (song) and the corresponding attribute set may look as follows. For better readability, we assume that music files are distinguished and identified by the song title.

$$\mathbb{I} = \{\textit{Music}, \textit{Impressive Instant}, \textit{Amazing}, \ldots\}$$

$$\mathbb{A} = \{\textit{Title}, \textit{Artist}, \textit{Album}, \textit{Comment}, \textit{Genre}, \textit{Year}\}$$

Using the data type set $\mathbb{D}$ from Example 1 the type ()-mapping looks like the following

$$\begin{aligned}
\text{type}\,(\mathit{Title}) &\mapsto (\mathtt{String}, \{\mathtt{==}\}) \\
\text{type}\,(\mathit{Artist}) &\mapsto (\mathtt{String}, \{\mathtt{==}\}) \\
\text{type}\,(\mathit{Album}) &\mapsto (\mathtt{String}, \{\mathtt{==}\}) \\
\text{type}\,(\mathit{Comment}) &\mapsto (\mathtt{String}, \{\mathtt{==}\}) \\
\text{type}\,(\mathit{Genre}) &\mapsto (\mathtt{int}, \{\mathtt{==,\ !=}\}) \\
\text{type}\,(\mathit{Year}) &\mapsto (\mathtt{Date}, \{\mathtt{<,\ >,\ ==,\ !=}\})
\end{aligned}$$

The value ()-mapping for the song *Amazing* is

$$\begin{aligned}
\text{value}\,(\mathit{Amazing}, \mathit{Title}) &\mapsto \mathtt{Amazing} \\
\text{value}\,(\mathit{Amazing}, \mathit{Artist}) &\mapsto \mathtt{Madonna} \\
\text{value}\,(\mathit{Amazing}, \mathit{Album}) &\mapsto \mathtt{Music} \\
\text{value}\,(\mathit{Amazing}, \mathit{Comment}) &\mapsto \mathtt{my\ favorite\ song} \\
\text{value}\,(\mathit{Amazing}, \mathit{Genre}) &\mapsto \mathtt{13} \\
\text{value}\,(\mathit{Amazing}, \mathit{Year}) &\mapsto \mathtt{2000}
\end{aligned}$$ □

Next, an *elementary filter* is defined. It specifies a constraint for a certain attribute. In addition, a matching function Δ_{EF} between an information item's attribute and an elementary filter is specified. On a successful match, the constraint given in the elementary filter is satisfied by the attribute.

Elementary filters serve as building blocks for an information wish that is in turn expressed by a complex filter (see Definition 11).

Definition 10 (Elementary Filter and Δ_{EF}-Function) An *elementary filter* φ is a tuple $(\alpha, \omega, \mathbf{v}) \in \mathbb{A} \times \text{operator}(\mathbb{D}) \times \mathbb{V}_{\mathbb{D}} \backslash \{\mathtt{undef}\}$ such that

$$\begin{aligned}
\omega &\in \Omega_\alpha \\
\mathbf{v} &\in \mathbf{S}_\alpha
\end{aligned}$$

where $\mathbf{S}_\alpha$ and Ω_α are defined by $(\mathbf{S}_\alpha, \Omega_\alpha) = \text{type}(\alpha)$.

Let $\varphi = (\alpha, \omega, \mathbf{v})$ be an elementary filter. Let π be an information item. We define a matching function between π and φ as follows

$$\Delta_{\text{EF}}(\pi, \varphi) := \begin{cases} \omega(\text{value}(\pi, \alpha), \mathbf{v}) : & \text{value}(\pi, \alpha) \neq \mathtt{undef} \\ \\ 0 : & \textit{otherwise} \end{cases}$$ □

On a match, the Δ_{EF}-function will yield 1 (true), otherwise it will yield 0 (false). If the value for an attribute is not known, it is set to `undef`. Applying an elementary filter to such an attribute will always yield 0. Figure 4.1 outlines the algorithm for Δ_{EF}-function in pseudo code.

For the sake of clarity a short example is given below.

```
/* Function: $\Delta_{\mathrm{EF}}(\pi, \varphi)$
 *    Matches an information item against
 *    an elementary filter
 * Parameters:
 *    $\pi$: an information item
 *    $\varphi = (\alpha, \omega, \mathbf{v})$: an elementary filter
 * Returns:
 *    TRUE  - the value for attribute $\alpha$ of information
 *            item $\pi$ fulfills the elementary filter $\varphi$
 *            constraint
 *
 *    FALSE - otherwise
 */
$\mathbf{v}_\alpha$ := value$(\pi, \alpha)$
if ($\mathbf{v}_\alpha \neq$ undef and $\omega(\mathbf{v}_\alpha, \mathbf{v}) = 1$) then
    return TRUE;
else
    return FALSE;
```

Figure 4.1: Algorithm outline for computing Δ_{EF}

Example 4 An elementary filter that matches all pop songs performed by the artist Madonna looks like the following

$$(\mathit{Artist}, ==, \texttt{Madonna})$$

Similarly, an elementary filter that matches all pop songs not belonging to the pop music genre[1] looks like the following

$$(\mathit{Genre}, !=, 13) \qquad \square$$

Proposition 1 *Let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model. For each $\pi \in \mathbb{I}$ and each $\alpha \in \mathbb{A}$ an elementary filter $\varphi^* = (\alpha, \omega^*, \mathbf{v}^*)$ with $\Delta_{\mathrm{EF}}(\pi, \varphi^*) = 1$ exists.* □

Proof Choose $\omega^* = \omega^{\mathrm{EQ}}$ and $\mathbf{v}^* = \text{value}(\pi, \alpha) \neq \texttt{undef}$. By Definition 10 and Definition 7 for ω^{EQ} it follows $\Delta_{\mathrm{EF}}(\pi, \varphi^*) = \omega^{\mathrm{EQ}}(\text{value}(\pi, \alpha), \mathbf{v}^*) = 1$. ■

Since an information item may be tagged with more than one attribute, it should be possible to apply several elementary filters on an information item. This would allow one to filter for *"songs from the artist 'Madonna' that were released before the year 2001."* This is achieved by combining elementary filters to a logical expression and is defined by a *complex filter* in the following way.

[1] According to the ID3v1 specification [NM05], the pop music genre is defined by the value 13.

Definition 11 (Complex Filter and Δ_{CF}-Function) Let $\mathbb{F}_0$ be the set of all elementary filters, and let $\mathfrak{B} = (\Sigma, \mathsf{V}, \mathsf{R}, \langle\texttt{complex filter}\rangle)$ be a Backus-Naur form with

$$\begin{aligned} \Sigma &= \{(,), \vee, \wedge, \neg\} \cup \mathbb{F}_0 \\ \mathsf{V} &= \{\langle\texttt{complex filter}\rangle, \langle\texttt{elementary filter}\rangle\} \end{aligned}$$

and the following production rules:

$$\begin{array}{lll} \mathsf{R} = \{ \ \langle\texttt{complex filter}\rangle & ::= \langle\texttt{elementary filter}\rangle & | \\ & \ \ \ \ \ \neg\langle\texttt{complex filter}\rangle & | \\ & \ \ \ \ \ (\langle\texttt{complex filter}\rangle \vee \langle\texttt{complex filter}\rangle) & | \\ & \ \ \ \ \ (\langle\texttt{complex filter}\rangle \wedge \langle\texttt{complex filter}\rangle) & , \\ \quad \langle\texttt{elementary filter}\rangle & ::= \varphi_i & \} \end{array}$$

with $\varphi_i \in \mathbb{F}_0$. We define a complex filter as an element out of the formal language $\mathsf{L}(\mathfrak{B})$, $\Phi \in \mathsf{L}(\mathfrak{B})$. The set of all complex filters is labeled with $\mathbb{F} := \mathsf{L}(\mathfrak{B})$.

Let Φ, Φ', Φ'' be complex filters. Let φ be an elementary filter. Let π be an information item. We define a matching function between π and Φ as follows

$$\Delta_{\mathrm{CF}}(\pi, \Phi) := \begin{cases} \Delta_{\mathrm{EF}}(\pi, \varphi) \quad \text{if} \quad \Phi \equiv \varphi \\ \neg\, \Delta_{\mathrm{CF}}(\pi, \varphi) \quad \text{if} \quad \Phi \equiv \neg\Phi' \\ \Delta_{\mathrm{CF}}(\pi, \Phi') \vee \Delta_{\mathrm{CF}}(\pi, \Phi'') \quad \text{if} \quad \Phi \equiv (\Phi' \vee \Phi'') \\ \Delta_{\mathrm{CF}}(\pi, \Phi') \wedge \Delta_{\mathrm{CF}}(\pi, \Phi'') \quad \text{if} \quad \Phi \equiv (\Phi' \wedge \Phi'') \end{cases}$$

□

In the Δ_{CF}-definition, the $\equiv$ sign expresses syntactical equivalence.

For an arbitrary information item π and a complex filter Φ, the Δ_{CF}-function is recursively defined and yields 1 (true), if all attributes of π fulfill related constraints, i.e., elementary filters, expressed in Φ. Otherwise Δ_{CF} returns 0 (false).

Proposition 2 *Let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model. Let $\pi \in \mathbb{I}$ be an information item arbitrarily tagged by the* value() *function. Let Φ be an arbitrarily complex filter for $\mathfrak{I}$. It holds $\Delta_{\mathrm{CF}}(\pi, \Phi) \in \mathbb{B} := \{0, 1\}$.* □

Proof For proving this assertion we use structural induction. Let $\varphi = (\alpha, \omega, \mathbf{v})$ be an elementary filter. By Definition 10, we see that $\Delta_{\mathrm{EF}}(\pi, \varphi) \in \{0, \omega(\text{value}(\pi, \alpha), \mathbf{v})\}$ and since by Definition 7 ω maps to $\{0, 1\}$, it follows that $\Delta_{\mathrm{EF}}(\pi, \varphi) \in \{0, 1\}$.

Now, every complex filter match function Δ_{CF} associates boolean values with boolean operators NOT, AND, and OR (see Definition 11). Therefore, by structural induction we are left with four possibilities:

$$\begin{array}{llll} \text{a.)} & \Delta_{\mathrm{CF}} & \equiv \Delta_{\mathrm{EF}}() & \in \mathbb{B} \\ \text{b.)} & \Delta_{\mathrm{CF}} & \equiv \neg\, \Delta_{\mathrm{CF}}() & \in \mathbb{B} \\ \text{c.)} & \Delta_{\mathrm{CF}} & \equiv \Delta_{\mathrm{CF}}() \vee \Delta_{\mathrm{CF}}() & \in \mathbb{B} \\ \text{d.)} & \Delta_{\mathrm{CF}} & \equiv \Delta_{\mathrm{CF}}() \wedge \Delta_{\mathrm{CF}}() & \in \mathbb{B} \end{array}$$

Finally, we conclude $\Delta_{\mathrm{CF}} \in \mathbb{B}$. ■

Theorem 1 *Let* $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \mathrm{type}, \mathrm{value})$ *be an information model. For each* $\pi \in \mathbb{I}$ *there exists a complex filter* Φ^* *with* $\Delta_{\mathrm{CF}}(\pi, \Phi^*) = 1$ □

Proof Let $\mathbb{A} = \{\alpha_1, \alpha_2, \dots \alpha_n\}$. By Proposition 1 there exist for each $\alpha_i \in \mathbb{A}$ an elementary filter $\varphi_i^* = (\alpha_i, \omega, \mathbf{v})$ with $\Delta_{\mathrm{EF}}(\pi, \varphi_i^*) = 1$. Now set $\Phi^* = ((\dots((\varphi_1^* \wedge \varphi_2^*) \wedge \varphi_i^*) \wedge \dots \wedge \varphi_n^*)$. With Definition 11 it follows $\Delta_{\mathrm{CF}}(\pi, \Phi^*) = 1$. ■

Example 5 An interest in songs from Madonna released before 2001 might be expressed by combining two elementary filters to one complex filter as shown below:

$$(\mathit{Artist}, ==, \texttt{Madonna}) \wedge (\mathit{Year}, <, \texttt{2001})$$

A complex filter also allows several constraints for the same attribute to be expressed. For example, an interest in any music file that was not released in the years 2000 and 2001 would look like

$$(\mathit{Year}, \,!\!=, \texttt{2001}) \wedge (\mathit{Year}, \,!\!=, \texttt{2001})$$

Since ¬ (NOT) is included in the complex filter definition as a unary operator, the following filter is equivalent

$$\neg(\mathit{Year}, ==, 2000) \wedge \neg(\mathit{Year}, ==, 2001)$$ □

From now on we simply write 'filter' instead of 'complex filter' if there is no ambiguity.
Figure 4.2 outlines the algorithm for Δ_{CF}.

Now, we are able to define a node profile (see page 37) that is composed of a set of information items and a set of complex filters. In addition, a match function Δ between these two sets is defined.

Definition 12 (Node Profile and Δ-Function) A node profile is defined as the following tuple

$$\mathbf{Profile} := (\mathbf{iHave}, \mathbf{iWish})$$

where **iHave** is a subset of the information item set, $\mathbf{iHave} \subseteq \mathbb{I}$, and **iWish** is a subset of the complex filter set, $\mathbf{iWish} \subseteq \mathbb{F}$.

Let $\mathbf{iHave}_{\mathrm{A}}$ be the set of information items offered by user A, let $\mathbf{iWish}_{\mathrm{B}}$ be the set of complex filters that express the interest of user B. We define the matching function between $\mathbf{iHave}_{\mathrm{A}}$ and $\mathbf{iWish}_{\mathrm{B}}$:

$$\Delta(\mathbf{iHave}_{\mathrm{A}}, \mathbf{iWish}_{\mathrm{B}}) := \{\pi \in \mathbf{iHave}_{\mathrm{A}} \mid \exists \Phi \in \mathbf{iWish}_{\mathrm{B}}, \Delta_{\mathrm{CF}}(\pi, \Phi) = 1\}$$ □

All songs in the next example are from the artist *Madonna*.

```
/* Function: $\Delta_{\mathrm{CF}}(\pi, \Phi)$
 *     Matches an information item against
 *     a complex filter
 * Parameters:
 *     $\pi$: an information item
 *     $\Phi$: a complex filter
 * Returns:
 *     TRUE  - the value for all attributes of information
 *             item $\pi$ fulfills the complex filter $\Phi$
 *             constraints
 *
 *     FALSE - otherwise
 */
switch($\Phi$)
  case $\Phi \equiv \varphi^*$:                              /* $\varphi^* \in \mathbb{F}_0$ */
     return $\Delta_{\mathrm{EF}}(\pi, \varphi^*)$;
  case $\Phi \equiv \neg\Phi'$:                             /* $\Phi' \in \mathbb{F}$ */
     return NOT($\Delta_{\mathrm{CF}}(\pi, \Phi')$);
  case $\Phi \equiv (\Phi' \vee \Phi'')$:                    /* $\Phi', \Phi'' \in \mathbb{F}$ */
     return ($\Delta_{\mathrm{CF}}(\pi, \Phi')$) OR ($\Delta_{\mathrm{CF}}(\pi, \Phi'')$);
  case $\Phi \equiv (\Phi' \wedge \Phi'')$:                  /* $\Phi', \Phi'' \in \mathbb{F}$ */
     return ($\Delta_{\mathrm{CF}}(\pi, \Phi')$) AND ($\Delta_{\mathrm{CF}}(\pi, \Phi'')$);
```

Figure 4.2: Algorithm outline for computing Δ_{CF}

Example 6 Let $\mathbf{Profile_A}$ be a user profile of user A with

$$\mathbf{iHave}_{\mathrm{A}} = \{\ \textit{Music, Amazing}\ \}$$
$$\mathbf{iWish}_{\mathrm{A}} = \{\ (\textit{Artist}, ==, \texttt{Madonna})\ \}$$

and $\mathbf{Profile_B}$ be a user profile of user B with

$$\mathbf{iHave}_{\mathrm{B}} = \{\ \textit{Impressive Instant}\ \}$$
$$\mathbf{iWish}_{\mathrm{B}} = \{\ (\textit{Artist}, ==, \texttt{Billy Idol})\ \}$$

Matching the two user profiles results in

$$\Delta(\mathbf{iHave}_{\mathrm{A}}, \mathbf{iWish}_{\mathrm{B}}) = \emptyset$$
$$\Delta(\mathbf{iHave}_{\mathrm{B}}, \mathbf{iWish}_{\mathrm{A}}) = \{\ \textit{Impressive Instant}\ \}$$

since all songs are from the artist *Madonna* and none from the artist *Billy Idol*. Therefore, user A would get the new song *Impressive Instant* from user B. User B would not receive any song from user A, as the distribution of songs in musicClouds takes place depending on the value of the Δ function. Figure 4.3 outlines the profile matching algorithms for Δ. □

```
/* Function: $\Delta(\textbf{iHave}_\text{A}, \textbf{iWish}_\text{B})$
 *    Matches an information have of user A with
 *    an information wish of user B
 * Parameters:
 *    $\textbf{iHave}_\text{A}$: an information have of user A
 *    $\textbf{iWish}_\text{B}$: an information wish of user B
 * Returns:
 *    A result set $\mathbf{R} \subseteq \textbf{iHave}_\text{A}$ with all information
 *    items from user A that match the information
 *    wish of user B.
 */
$\mathbf{R} := \emptyset$;
foreach $\pi \in \textbf{iHave}_\text{A}$ do
   foreach $\Phi \in \textbf{iWish}_\text{B}$ do
      if ($\Delta_{\text{CF}}(\pi, \Phi) = \text{TRUE}$)
        $\mathbf{R} := \mathbf{R} \cup \{\pi\}$
        break;
      endif;
   done;
done;
return R;
```

Figure 4.3: Algorithm outline for computing Δ

4.2 Modeling Category, Date and Location Information

Chapter 3 discussed the importance and usefulness of category information within opportunistic network applications (see Section 3.5.1). In addition, date and location information often needs to incorporated into an information model. For example, *proximity based* services ask for location and date information to be useful. Thus, this section presents a simple solution how this data can be modeled uniformly as attributes with appropriate data types. Since there might be other suitable models for category, date and especially location information, the presented solutions are called attribute proposals.

4.2.1 Category

By category we understand a division of information items within a system of classification within an application domain. A category attribute allows information items to be grouped. This simplifies the task of displaying or browsing information items and is a common approach.

Attribute Proposal 1 (Category) Let $\Im = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model. Let $\mathbb{C}$ be a non-empty set of categories. A particular attribute $\alpha_0 \in \mathbb{A}$ is called *category attribute* if the following holds:

$$
\begin{aligned}
&D_0 := (\mathbf{S}_0, \Omega_0) \in \mathbb{D} \;\text{ satisfies }\; \mathbf{S}_0 = \mathcal{P}(\mathbb{C}) \setminus \{\emptyset\} && \text{(i)}\\
&\text{type}(\alpha_0) \mapsto D_0 && \text{(ii)}\\
&\text{value}(\pi, \alpha_0) \neq \texttt{undef} \;\text{ for all }\; \pi \in \mathbb{I} && \text{(iii)}\\
&\Omega_0 = \{\omega_0^{\mathrm{OR}}, \omega_0^{\mathrm{AND}}, \omega_0^{\mathrm{EQ}}\} && \text{(iv)}
\end{aligned}
$$

For $x, y \in \mathbf{S}_0$, the binary operators are

$$\omega_0^{\mathrm{OR}}(x,y) := \begin{cases} 1: & x \cap y \neq \emptyset \\ 0: & \textit{otherwise} \end{cases}$$

$$\omega_0^{\mathrm{AND}}(x,y) := \begin{cases} 1: & x \supseteq y \\ 0: & \textit{otherwise} \end{cases}$$

$$\omega_0^{\mathrm{EQ}}(x,y) := \begin{cases} 1: & x = y \\ 0: & \textit{otherwise} \end{cases}$$

As stated in condition (i) and (ii), possible values for α_0 are out of the set $\mathcal{P}(\mathbb{C}) \setminus \{\emptyset\}$. This allows us to map an information item π to more than one category.

Condition (iii) demands a proper assignment to one or more categories for each information item.

The special set of operators, as expressed in condition (iv), allows a user to express a filter on the category attribute in a very flexible manner. We illustrate this in the following example, again based on the musicClouds application.

Example 7 The *Genre* attribute serves as a natural candidate for the category attribute. The application knows the following genres:

$$\mathbb{C} = \{\text{Pop}, \text{Rock}, \text{Folk}, \text{Jazz}\}$$

The singer and songwriter *Katie Melua* combines the Pop with the Jazz genre in her songs. Thus, for example, her song *Call Off The Search* would belong to both categories. Expressed in the model:

$$
\begin{aligned}
&\text{type}(\textit{Genre}) \mapsto (\,\mathcal{P}(\{\mathbb{C}\}) \setminus \{\emptyset\},\, \{\omega_0^{\mathrm{OR}}, \omega_0^{\mathrm{AND}}, \omega_0^{\mathrm{EQ}}\}\,)\\
&\text{value}(\textit{Call Off The Search}, \textit{Genre}) \mapsto \{\text{Pop}, \text{Jazz}\}
\end{aligned}
$$

Next, a filter on the genre category attribute may choose between ω_0^{OR}, ω_0^{AND}, and

ω_0^{EQ} operator, depending on what kind of interest should be expressed.

$$(\mathit{Genre},\ \omega_0^{\mathrm{OR}},\ \{\mathrm{Pop}, \mathrm{Folk}\}) \tag{a}$$

$$(\mathit{Genre},\ \omega_0^{\mathrm{AND}},\ \{\mathrm{Pop}, \mathrm{Folk}\}) \tag{b}$$

$$(\mathit{Genre},\ \omega_0^{\mathrm{EQ}},\ \{\mathrm{Pop}, \mathrm{Jazz}\}) \tag{c}$$

□

Filter (a) expresses an interest in songs belonging to the Pop music *or* Folk music genre. This filter would successfully match against *Melua*'s song. Filter (b) expresses interest in music that simultaneously belongs to the Pop and Folk genre. A match would fail here, since the song is not a Folk song. Note that some other song belonging to Pop, Folk, and Jazz music would match (see the $\supseteq$ operator in the definition). The most accurate wish is expressed in filter (c). This filter expresses interest in songs that belong exactly to the two genres: Pop *and* Jazz music.

4.2.2 Time

The next attribute models time in the form of an interval.

Attribute Proposal 2 (Time Interval) Let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \mathrm{type}, \mathrm{value})$ be an information model. Let a data type

$$D^* = (\mathbf{S}^*, \Omega^*) := (\mathtt{Date}, \{\mathtt{<},\ \mathtt{>},\ \mathtt{==},\ \mathtt{!=}\}) \in \mathbb{D}$$

model a point in time. The data type for a *time interval attribute* $\alpha_1 \in \mathbb{A}$ might be modeled as follows[2]:

$$\mathbf{S}_1 := \{(t^-, t^+) \mid t^-, t^+ \in \mathbf{S}^* \text{ and } \mathtt{before}(t^-, t^+) = 1\} \tag{i}$$

$$\Omega_1 := \{\omega_1^{\cap}\} \tag{ii}$$

$$D_1 := (\mathbf{S}_1, \Omega_1) \tag{iii}$$

$$\mathrm{type}\,(\alpha_1) \mapsto D_1 \tag{iv}$$

with the operator $\omega_1^{\cap}$ defined as follows

$$\omega_2^{\cap}(x, y) := \begin{cases} 1: & \text{the intervals } x \text{ and } y \text{ have a common point in time} \\ 0: & \textit{otherwise} \end{cases}$$

Equation (i) – (iii) define a data type D_1 that models time intervals. Statement (iv) maps an attribute to D_1. An algorithm outline for operator $\omega_2^{\cap}$ is given in Figure 4.4. For a time interval $t = (t^-, t^+) \in \mathbf{S}_1$, the algorithm uses two restriction mappings start () and end () as follows

$$\mathrm{start}: \mathbf{S}_1 \to \mathbf{S}^*, (t^-, t^+) \mapsto t^-$$

$$\mathrm{end}: \mathbf{S}_1 \to \mathbf{S}^*, (t^-, t^+) \mapsto t^+$$

[2] The $\mathtt{before}(t^-, t^+)$ function yields 1 (true), if t^- lies before t^+ in time.

```
/* Function: matchTimeIntervals($t$, $s$)
 *     Implements $\omega_2^{\cap}()$. Tests if time interval $t$
 *     overlaps with time interval $s$
 *
 * Parameters:
 *     $t$: a time interval
 *     $s$: a time interval
 * Returns:
 *     TRUE   - time intervals overlap
 *
 *     FALSE  - otherwise
 */
$t^- :=$ start($t$);  $t^+ :=$ end($t$);
$s^- :=$ start($s$);  $s^+ :=$ end($s$);

if ($t^- < s^-$) and ($t^+ > s^-$) then
    return TRUE;
else if ($s^- < t^-$) and ($s^+ > t^-$) then
    return TRUE;
else
    return FALSE;
```

Figure 4.4: Algorithm outline for matching two time intervals

4.2.2.1 Current Time Adjustment

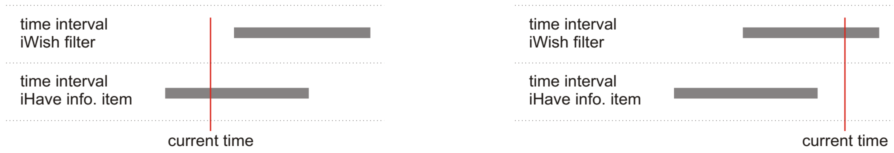
There may be situations when it is useful to modify a time interval of a filter. For example, an opportunistic network applications might allow information wishes to be expressed restricted to a time interval. An example filter that matches a pop concert event from June to September looks like the following:

$$(\mathit{Event}, ==, \texttt{Pop Concert}) \wedge (\mathit{Period}, \omega_2^{\cap}, (\texttt{JUN, SEP})) \tag{i}$$

A pop concert event π that takes place in July is tagged

$$\begin{aligned} \text{value}\,(\pi, \mathit{Event}) &\mapsto \texttt{Pop Concert} \\ \text{value}\,(\pi, \mathit{Period}) &\mapsto \texttt{(JUL, JUL)} \end{aligned} \tag{ii}$$

Now suppose a user A has filter (i) on his iWish-list and another user B offers information (ii) on his iHave-list. If user A and user B meet, for example, in July (Figure 4.5(b)), there is no problem. The information is successfully matched and useful for user A. But, suppose, user A and user B meet in August (see Figure 4.5(b)). The wish (i) is still valid, since it lasts until September. But the information item (ii) is not useful, since the concert has already happened. One solution would be to remove (ii) from user B's iHave-list. But what if he wants to keep it?



(a) ... no difference: matching succeeds

(b) ... a difference: matching fails

Figure 4.5: Current time makes ...

Another approach is to take the current time into account and adjust the filter (i) if the user wishes. The application might adjust the interval start for an interval $i = (t^-, t^+)$ according to the following formula:

$$t^- := max(t^-, \text{current time})$$

If this formula leads to an invalid time interval, i.e., $t^- > t^+$, the application might consider removing the filter entirely, since it will not match any useful information any more.

In conclusion, it should be under user control, if the current time is taken into account or not. Some users might be interested in outdated events, others not.

4.2.3 Location

Location information is needed for applications that share knowledge about physical locations, for example information about cultural events, as well as for Information Sprinklers that provide location based information and services. The next attribute proposal presents a simple approach. A location is modeled by a unique identifier, for example, a character string, and represents only a very rough image of the real world. For example, city names or zip codes might serve as identifiers.

All system-known locations are grouped together in a set. Using a powerset, it is possible to assign several locations to a location attribute. This allows us to express information like *"The Rhein-Main Pop Festival happens on July, 2nd, 2006, in Mainz* ***and*** *Wiesbaden"*.

Attribute Proposal 3 (Location) Let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model. Let $\mathbb{L}$ be a set of locations. A particular attribute $\alpha_L \in \mathbb{A}$ for tagging an information item with location information might be modeled as follows.

$$D_0 := (\mathbf{S}_0, \Omega_0) \in \mathbb{D} \;\; \text{satisfies} \;\; \mathbf{S}_0 = \mathcal{P}(\mathbb{L}) \setminus \{\emptyset\} \tag{i}$$

$$\text{type}(\alpha_L) \mapsto D_0 \tag{ii}$$

$$\Omega_0 = \{\omega_0^{\text{OR}}, \omega_0^{\text{AND}}, \omega_0^{\text{EQ}}\} \tag{iii}$$

with the following operators

$$\omega_0^{\mathrm{OR}}(x,y) := \begin{cases} 1: & x \cap y \neq \emptyset \\ 0: & otherwise \end{cases}$$

$$\omega_0^{\mathrm{AND}}(x,y) := \begin{cases} 1: & x \supseteq y \\ 0: & otherwise \end{cases}$$

$$\omega_0^{\mathrm{EQ}}(x,y) := \begin{cases} 1: & x = y \\ 0: & otherwise \end{cases}$$

A location attribute is modeled very similarly to the category attribute, with the exception that we allow a location attribute value to be equal to `undef`, if the information is not available.

Example 8 An application might use city names to tag events with location information. An example location set might be

$$\mathbb{L} = \{\text{Darmstadt}, \text{Frankfurt}, \text{Mainz}, \text{Wiesbaden}\}$$

Using the data type definition of attribute proposal 3 on page 65, a location attribute allocation for a festival that takes place in Wiesbaden and Mainz looks like the following:

$$\begin{aligned} &\text{type}(\mathit{Location}) \mapsto (\mathcal{P}(\{\mathbb{L}\}) \setminus \{\emptyset\}, \{\omega_0^{\mathrm{OR}}, \omega_0^{\mathrm{AND}}, \omega_0^{\mathrm{EQ}}\}) \\ &\text{value}(\mathit{Rhein\text{-}Main\ Pop\ Festival}, \mathit{Location}) \mapsto \{\text{Wiesbaden}, \text{Mainz}\} \end{aligned}$$

Again, an elementary filter on the location attribute may choose between ω_0^{OR}, ω_0^{AND}, and ω_0^{EQ} operator, depending on what kind of location interest should be expressed.

$$(\mathit{Location},\ \omega_0^{\mathrm{OR}},\ \{\text{Wiesbaden}, \text{Mainz}\}) \tag{a}$$

$$(\mathit{Location},\ \omega_0^{\mathrm{AND}},\ \{\text{Wiesbaden}, \text{Mainz}\}) \tag{b}$$

$$(\mathit{Location},\ \omega_0^{\mathrm{EQ}},\ \{\text{Wiesbaden}, \text{Mainz}\}) \tag{c}$$

Filter (a) expresses interest in events that happen in Wiesbaden **or** Mainz. Filter (b) matches only events that happen in both cities (but maybe more). Finally, filter (c) takes only events into account that happen in Wiesbaden **and** Mainz, nothing more, nothing less. □

As already stated, this location model is very simple. Applications with the need for a more sophisticated location representation need to implement suitable location data types to fit into the proposed model.

4.2.4 Limitation

Recall that the semantics of an information item that is tagged with several attributes is defined by *and*-semantics. We give an example.

Let $\mathfrak{I} = (\mathbb{I}, \mathbb{A}, \mathbb{D}, \text{type}, \text{value})$ be an information model with $\mathbb{A} = \{\alpha_0, \alpha_1\}$ and $\mathbb{D} = \{D_0, D_1\}$, let $\pi_0 \in \mathbb{I}$ be an information item. Furthermore, the following holds

$$\begin{aligned}\text{type}(\alpha_0) &\mapsto D_0 = (\mathcal{P}(\mathbf{S}_0), \Omega_0)\\ \text{type}(\alpha_1) &\mapsto D_1 = (\mathcal{P}(\mathbf{S}_1), \Omega_1)\\ \text{value}(\pi_0, \alpha_0) &\mapsto d0_i, \text{ with } d0_i \in \text{domain}(D_0)\\ \text{value}(\pi_0, \alpha_1) &\mapsto d1_j, \text{ with } d1_j \in \text{domain}(D_1)\end{aligned}$$

The semantics are: $\alpha_0 = d0_i$ **and** $\alpha_1 = d1_j$ are true for information item π_0. Let $d0_i$ and $d1_j$ be elements out of power sets. Thus they are sets themselves, for example $d0_i = \{s0_{i_l}, s0_{i_k}\}$ and $d1_j = \{s1_{j_m}, s1_{j_n}\}$. The model is not able to relate individual elements out of $d0_i$ to individual elements out of $d1_j$; in other words, it cannot build tuples (x, y) with $x \in d0_i$ and $y \in d1_j$. Thus, an information item π that models a *moving exhibition event* that charges different entry fees at different locations, for example asking for (small fee, village) and (big fee, city) tuples, is not possible. In order to model this kind of applications there are two options.

(1) The application might split the information item into several items and tag each with a (fee, location)-pair.

(2) The semantics have to be implemented in a custom data type. This data type must fulfill Definition 7.

4.3 Application Specification and Design

Using the formal information model as a basis, parts of the design and specification task of an opportunistic network application can be broken down into four steps. These steps follow a *top-down* approach as illustrated in Figure 4.6. The distinct steps serve the following purpose:

(1) **Application Description**

The purpose of the application is described in a couple of sentences. This step includes design choices as described in Section 4.3.1. In addition, hardware requirements are specified. For instance, the minimal amount of application memory to be available or which wireless communication technology to use is specified.

(2) **Information Item Specification**

Since the core functionality of an opportunistic network application is to disseminate information items among users, a single information item is

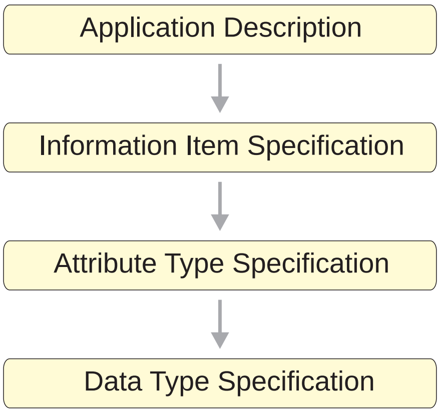


Figure 4.6: Design steps

specified in this step. This includes the definition of an information item's internal storage representation or format, for example, the file format for music files and the selection of appropriate attributes to tag an information item adequately.

(3) **Attribute Type Specification**

For each attribute the type is specified in this step. This influences the next step. For example, the musicClouds application uses the ID3 tags [NM05] as tag attributes.

(4) **Data Type Specification**

Having identified suitable attributes for the application, adequate data types that fulfill Definition 7 have to be specified in this last step.

The outcome of this step eases the programming language choice, since different programming languages offer different built-in data types or are shipped with convenient libraries, for example the Java collection classes [Sun02]. In general, a programming language that reduces the programming task is preferable.

4.3.1 Further Design Choices

Push vs. Pull: Chapter 3 discussed different choices for the data sharing protocol, namely a *push* or *pull* model (see Section 3.3.2), with their respective advantages and disadvantages respectively. In the application design phase, the most suitable protocol is specified, either *push* or *pull* or even a combination of both.

User Notification: As discussed in Chapter 2, the spectrum for opportunistic network applications ranges from *active collaboration* to *passive collaboration*. Active collaboration requires a feedback loop to the user in case of a match, whereas passive collaboration goes without. According to the main purpose of the application, the design phase has to choose between the following:

(1) User notification on every successful match and information exchange.

(2) User notification on a per iWish-list entry basis, i.e., some matches result in a user notification, while others not. Therefore, the application needs to store this additional information per iWish-list entry.

(3) No user notification on a successful match. This choice was implemented in the adPASS prototype.

Privacy Preservation: The next chapter explains a mechanism for preserving the privacy of a user by avoiding static data on all network layers and within the application. Depending on the purpose of the opportunistic network application, a designer has to specify whether privacy preservation is important and supported by the application or deliberately turned off to increase the usefulness of the application. For example, for purely passive collaboration applications that disseminate personal data, privacy preservation is crucial, whereas people finder-like applications need to reveal a user's identity to be useful. Thus, privacy preservation plays a secondary role only.

4.4 Using the Model

Having introduced the formal information model, this section presents selective source code excerpts from musicClouds. This illustrates a simple and straightforward model implementation in the Java programming language for one of our opportunistic network prototypes. We use the symbol $\boxed{\cdots}$ within a source code listing to indicate that discussion-irrelevant code is omitted.

The musicClouds prototype allows users to share mp3 encoded music files in a spontaneous manner. Interest in music genres and shared music is specified in the node profile (see page 59). The iHave part consists of `InfoItemMp3` entries and music interests (iWish part) are expressed by using `ComplexFilter` entries. For more details on musicClouds, see Section 6.1.2.

The next three listings 4.1, 4.2, and 4.3 show data type implementations according to Definition 7. The `MyString` data type is used for title name, artist name, album name, and a user comment of a mp3 file. `MyGenre` represents a certain music genre and `MyRating` allows users to express their opinion on a song.

```java
package musicClouds.InfoModel;

```

```
public class MyString {
    public static final int OP_EQUALS     = 1;
    public static final int OP_STARTSWITH = 2;

    private String value = null;

    public MyString(String v) {  ···

    /** operators */
    boolean equals(MyString other) { {   ···
    boolean startsWith(MyString other) {  ···

    public String getValue() {  ···
    public void setValue(String value) {  ···
    public String toString() {  ···

}
```

Listing 4.1: Source `MyString` data type

```
package musicClouds.InfoModel;

public class MyGenre {
    public static final int OP_EQUALS    = 1;
    public static final int OP_MATCHONE  = 2;
    public static final int OP_MATCHALL  = 3;

    public static final int maxNumberCategories = 81;
    public static final int BLUES        =  0;
    public static final int CLASSIC_ROCK =  1;
    public static final int COUNTRY      =  2;
     ···
    public static final int  FOLK        =  80;
     ···
    /** operators */
    public boolean equals(MyGenre other) {  ···
    public boolean matchOne (MyGenre other) {  ···
    public boolean matchAll(MyGenre other) {  ···

    public void setGenre(int genreId) {  ···
    public boolean[] getGenres() {  ···
    public boolean[] getValue() {  ···
    public String toString() {  ···
}
```

Listing 4.2: Source `MyGenre` data type

```
1  package musicClouds.InfoModel;
2
3  public class MyRating {
4          public static final int OP_EQUALS     = 1;
5          public static final int OP_BETTER     = 2;
6
7          int value = 0;
8
9          public MyRating(int v) {  …
10
11         /** operators */
12         boolean equals(MyRating other) {
13             if (value == other.getValue()) return true;
14             else return false;
15         }
16
17         boolean better(MyRating other) {
18             if (other == null) return false;
19
20             if (value < other.getValue()) return true;
21             else return false;
22         }
23
24         public int getValue() {  …
25         public void setValue(int value) {  …
26         public String toString() {  …
27 }
```

Listing 4.3: Source `MyRating` data type

Each data type declares a number of public accessible integer constants that represent supported operators. These constants are important for the elementary filter construction (see line 4 in listing 4.7). As required by Definition 7, each data type declares a set of binary operators which can be applied on a data type element and yield true or false. The number and semantics of these operators are defined by the application developer and affect the filter expressiveness. For example, if `MyRating` does not offer the operator `better` (see line 17 in listing 4.3) a user could not easily express a lower bound for a song rating.

With the data types at hand, the source code of an information item for an mp3 music file is shown in listing 4.4. Similar to the data type implementation, for each information item attribute a public accessible integer constant is declared (line 6 – line 11). These constants help specify the information item attribute an elementary filter is applied to (see line 3 in listing 4.7).

```
1  package musicClouds.InfoModel;
2  import java.io.File;
3
```

```
public class InfoItemMp3 {

    public static final int TITLE   = 1;
    public static final int ARTIST  = 2;
    public static final int ALBUM   = 3;
    public static final int RATING  = 4;
    public static final int COMMENT = 5;
    public static final int GENRE   = 6;

    private File     mp3     = null;
    private MyString title   = null;
    private MyString artist  = null;
    private MyString album   = null;
    private MyRating rating  = null;
    private MyString comment = null;
    private MyGenre  genre   = null;

    public MyString getAlbum() { ···
    public void setAlbum(MyString album) { ···
    public MyString getComment() { ···
    public void setComment(MyString comment) { ···
    public MyRating getRating() { ···
    public void setRating(MyRating rating) { ···
    public MyGenre getGenre() { ···
    public void setGenre(MyGenre genre) { ···
    public MyString getTitle() { ···
    public void setTitle(MyString title) { ···
    public MyString getArtist() { ···
    public void setArtist(MyString artist) { ···
    public File getMp3() { ···
    public void setMp3(File mp3) { ···
    public String toString() { ···
}
```

Listing 4.4: Source InfoItemMp3

Listing 4.5 shows the Java source code for an elementary filter φ, including the $\Delta_{\mathrm{EF}}(\pi, \varphi)$ matching function for an information item π applied to this elementary filter (`match()`). Line 4 – line 6 reflect the elementary filter Definition 10, i.e., an elementary filter consists of a `(attribute,operator,value)`-tuple.

```
1  package musicClouds.InfoModel;
2
3  public class ElementaryFilter {
4      private int attribute = 0;
5      private int operator  = 0;
6      private Object value  = null;
7
8      public ElementaryFilter(int attribute, int operator,
9                               Object value) { ···
10
11      public boolean match(InfoItemMp3 item) {
12          boolean result = false;
13
14          switch( this.attribute ) {
15          ···
16          case InfoItemMp3.RATING:
17              switch( this.operator ) {
18              case MyRating.OP_EQUALS:
19                  result = ((MyRating) value).equals(
20                                        item.getRating());
21                  break;
22              case MyRating.OP_BETTER:
23                  result = ((MyRating) value).better(
24                                        item.getRating());
25                  break;
26              }
27          ···
28          case InfoItemMp3.GENRE:
29              switch( this.operator ) {
30              case MyGenre.OP_EQUALS:
31                  result = ((MyGenre) value).equals(
32                                       item.getGenre());
33                  break;
34              case MyGenre.OP_MATCHONE:
35                  result = ((MyGenre) value).matchOne(
36                                       item.getGenre());
37                  break;
38              case MyGenre.OP_MATCHALL:
39                  result = ((MyGenre) value).matchAll(
40                                       item.getGenre());
41                  break;
42              }
43          ···
44          return result;
45      }
46
47      public String toString() { ···
48      public String lookup(int attribute, int operator) { ···
49  }
```

Listing 4.5: Source ElementaryFilter

Carrying out a matching is a two-stage process (for example, see line 28 – line 42). First, the information item attribute under observation is aligned with the attribute specified in the elementary filter. Second, with the use of the operator integer constant, the appropriate operator method is called in order to compare the elementary filter value with the attribute value. The result yields true or false.

```
package musicClouds.InfoModel;

public class ComplexFilter {

    public static final int TYPE_EL        = 1;
    public static final int TYPE_N_CF      = 2;
    public static final int TYPE_CF_AND_CF = 3;
    public static final int TYPE_CF_OR_CF  = 4;
    public static final int TYPE_unknown   = 5;

    private ComplexFilter left  = null;
    private ComplexFilter right = null;

    private ElementaryFilter el = null;
    private int type = ComplexFilter.TYPE_unknown;

    private ComplexFilter() { ⋯

    /** public constructors */
    public ComplexFilter(ElementaryFilter el) { ⋯
    public ComplexFilter(ComplexFilter cf, int type) { ⋯
    public ComplexFilter(ComplexFilter l_cf,
                         ComplexFilter r_cf,
                         int type) { ⋯

    /** matching */
    public  boolean match(InfoItemMp3 item) {
        boolean result = false;

        switch(type) {
        case ComplexFilter.TYPE_EL:
            result = el.match(item);
            break;
        case ComplexFilter.TYPE_N_CF:
            result = !(left.match(item));
            break;
        case ComplexFilter.TYPE_CF_AND_CF:
            result = (left.match(item) & right.match(item));
            break;
        case ComplexFilter.TYPE_CF_OR_CF:
            result = (left.match(item) | right.match(item));
            break;
        }
        return result;
    }
    public String toString() { ⋯
}
```

Listing 4.6: Source ComplexFilter

Listing 4.6 shows the source code for a complex filter (see Definition 11) together with the function `match()` (starting at line 27) that implements the $\Delta_{\mathrm{CF}}(\pi, \Phi)$-function. The Backus-Naur production rules are mapped to different constructors (line 20 – line 24) that allow complex and elementary filters to be combined recursively to a new complex filter. The internal structure of a complex filter is a binary tree. The branches are complex or elementary filters themselves and an integer attribute `type` records how to logically (not,and,or) evaluate the branches. (see line 30 – line 43).

```
1  ...
2  ElementaryFilter m = new ElementaryFilter(
3                               InfoItemMp3.ARTIST,
4                               MyString.OP_EQUALS,
5                               new MyString("Madonna"));
6
7  MyGenre genreInterests = new MyGenre();
8  genreInterests.setGenre(MyGenre.ROCK);
9  genreInterests.setGenre(MyGenre.POP);
10 ElementaryFilter g = new ElementaryFilter(
11                                InfoItemMp3.GENRE,
12                                MyGenre.OP_MATCHONE,
13                                genreInterests);
14
15 ComplexFilter madonna = ComplexFilter(m);
16 ComplexFilter rockOrPop = new ComplexFilter(g);
17 ComplexFilter cf = new ComplexFilter(madonna, rockOrPop,
18                      ComplexFilter.TYPE_CF_OR_CF);
19 ...
```

Listing 4.7: Source snippet using `ElementaryFilter` and `ComplexFilter`

The final listing 4.7 gives a short example of how to use the previously presented elementary and complex filter classes. At line 2 – line 5, an elementary filter that matches the artist name ”Madonna” is constructed. Then, line 7 – line 13, a second elementary filter that matches the genre ROCK or the genre POP is constructed. These two filters are combined to a complex filter `cf` at line 17. Altogether, the complex filter expresses interest into the following mp3 titles:

$$(Artist, ==, \texttt{Madonna}) \vee (Genre, \in, \{\texttt{ROCK,POP}\})$$

4.5 Summary

This chapter introduced a formalism for the data dissemination task in opportunistic networks. On the one hand, it covered a generic way to model the information to be disseminated with the notion of an Information Model (Definition 9). On the other hand, a node’s expression of an interest in some information was realized by means of a complex filter (Definition 11). With the formalism, we were able to outline basic matching algorithms in a programming language-independent fashion.

Furthermore, the formal modeling helps opportunistic network application designers and developers in their work with a simple top-down design methodology that eases the implementation task.

Finally, in Section 4.4, concrete examples were presented of how the model was implemented in the musicClouds application using the Java programming language.

Chapter 5

Acceptability

Opportunistic networks are formed by individual users and their devices. The data dissemination process is based upon the users' will to contribute to and participate in the network. A user

i) shares personal information. Knowledge and possession is shared in the form of entries in the iHave-list and interests are shared in the form of entries in the iWish-list.

ii) shares his device resources with the network in the form of memory, CPU, and battery power, by running an opportunistic network application.

Both issues may affect the acceptability of an application. The first issue puts user privacy at risk. Another user in communication range learns about the interests and knowledge of others. The second issue may reduce the utility of the device, since battery power might be drained or available memory might become low. Thus personal usability of a device may decrease. Consequently, user acceptability of opportunistic networks and its applications might be affected.

This chapter addresses both personal information sharing and device resource sharing with regards to privacy issues and user incentives respectively, in order to increase user acceptability. Section 5.1 presents a solution for preserving user privacy. This solution exploits the *one-hop* communication property of opportunistic networks. In Section 5.2, an incentive scheme for opportunistic networks is modeled.

The incentive scheme, together with the privacy preserving mechanism is applicable to a number of opportunistic network applications as well as other systems with similar communication patterns. We will give two examples in the last part of this chapter. See Appendix A for used abbreviations.

5.1 Preserving User Privacy

Users give away some personal information by putting items onto their iLists. In order to address a user's concerns about privacy issues and increase an application's

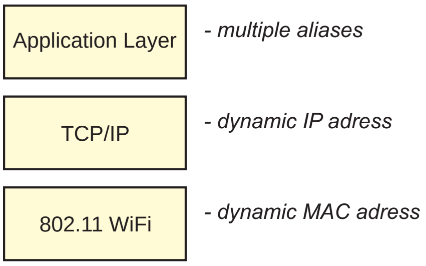


Figure 5.1: Dynamic IDs in a typical network stack

acceptance, one goal is to prevent other parties from constructing detailed user profiles simply by taking part in the network. The creation of a user profile that breaches a user's privacy needs to unambiguously map gathered information, in this sense, entries on an iHave-list and an iWish-list, to a user. Therefore, an attacker needs to learn about the unique identifier of a user that does not change over time. Suitable candidates to serve as a unique identifier are all pieces of static information that occur during communication. Examples in the widespread TCP/IP protocol suite are MAC addresses, IP addresses, and user names (on the application layer). For this reason, the core concept to preserve privacy in opportunistic networks is summarized in the following method:

> The utilization of dynamic and self-generated identifiers and aliases in each and every communication layer within an opportunistic network application.

This approach, illustrated in Figure 5.1, is feasible, since opportunistic networks deliberately do not support any message routing functionality and an individual node covers only a limited physical area. Therefore, a node is able to randomly change its identifiers and application aliases from time to time without doing any harm to the communication functionality.

On the application layer, we use public keys out of a number of cryptographic key pairs as aliases. A *multiple aliases* support on the application layer makes a series of communication actions issued by one node unlinkable to an attacker.

A cryptographic key pair consists of a private key and the corresponding public key. Technically, the alias is a byte sequence (modulus) derived from the public key. Figure 5.2 depicts the extracted modulus of a 512 bit RSA public key. Since this byte sequence represents the product of two randomly chosen prime numbers, this sequence is unique with a high probability and suitable to serve as a unique user alias.

In this thesis, we use private keys to digitally sign a message in order to protect its authenticity and integrity. Digital signatures can be verified by anyone who

```
00:c2:54:a1:ef:07:58:c9:91:d8:f3:a5:af:b9:53:
63:c7:a7:47:49:3d:a5:7e:65:c2:e6:ab:ed:6b:2e:
ef:36:0d:53:01:6e:e2:76:41:4b:54:89:8d:1a:9a:
d3:03:50:a0:c2:b0:1c:78:ba:38:81:fc:9c:ce:91:
0e:e1:ec:01:5b
```

Figure 5.2: Extracted RSA modulus (512) bit of a public key

knows the pubic key. In this sense, the cryptographic operations associated with the two keys are inverse to each other. We use the notion (P^+, P^-) to denote a public key P^+ and the matching private key P^-. As the computation of P^- out of P^+ is computationally infeasible in modern cryptographic schemes, P^+ can be made public. This property is exploited to let a user prove that he is the owner of a certain alias P^+.

The key pairs are generated on the device itself. Each user can change his alias as often he likes, since he can create a new key pair on his own. The strongest variant would be to use a new key pair for each communication. A more practical idea is to store a small set of key pairs and choose one alias at random for each information exchange. For this purpose, each node keeps a number of n self generated key pairs in a so-called *key bag* on the device:

$$KB := \{(P^+, P^-)_i \mid 1 \leq i \leq n\} \tag{5.1}$$

In addition, the use of public keys as aliases allows a user to prove his participation in previous communication. This property is exploited in the incentive scheme described in Section 5.2.

The concept of dynamic identifiers might not be enough in networks of small size. The worst case is a network consisting of two nodes only, with node A being an attacker who wants to create a profile of the other node. Node B, changing its identifiers from time to time, is still recognizable by A, since there is no other node around. Thus as a second means, we allow a node to specify a minimal number k of nodes in communication range, before the node takes part in the information exchange protocol.

Figure 5.3 illustrates this strategy in pseudo code. This mechanism of *cloaking* was proposed by Gruteser and Grunwald [GG03] and aims to conceal a user within a group of k people. In their work, a user is defined to be *k-anonymous* if and only if he is indistinguishable from at least $k - 1$ other users. Gruteser and Grunwald argue that a reasonable value for k is between 5 and 10.

5.1.1 Discussion

Our proposed combination of dynamic identifiers together with *cloaking* does not make profile creation impossible. The following attacks are possible:

```
activeNodes := 0;
for each node $\in$ neighborhood\_list do
  if (node is alive) then
    activeNodes++;
  endif
done
if (activeNodes > $k-1$) then
  /* start information exchange protocol */
else
  /* stay silent */
endif
```

Figure 5.3: *k*-anonymity strategy

1) An attacker carries out the Sybil attack [Dou02]. Here, a node presents multiple identities to the network. For this purpose, one device changes its ID frequently.

2) An attacker carries a bag with multiple devices (and thus multiple IDs) with him.

Both attacks are a threat, since the victim would erroneously assume that a sufficiently number of users were nearby. However, since the algorithm in Figure 5.3 tests the active nodes beforehand, an attacker's device must be able to change its IDs very quickly. At least, this might be difficult for the Sybil attack.

The following requirements need to be met by both attacks in order to be successful:

1) Any attack requires the physical presence of the attacker in the victim's vicinity, due to the limited communication range property in opportunistic networks.

2) The attacker needs a rough prediction of the victim's physical movements, in order to carry out the attack and not lose connection to the victim.

Altogether, if we weigh up the attack requirements with the potential outcome, i.e., the profile creation of one user, the remaining risk is negligible. Needless to say, a successful attack still needs to link the constructed profile to the victim's identity or name in reality, before the profile becomes useful, for example, for a target advertising company.

5.2 Incentive Scheme

For the sake of completeness, the following description slightly overlaps with Section 3.6.2. There we already sketched the basic idea, roles, and interaction pattern of an incentive scheme for opportunistic networks that aims to increase user acceptability. This section describes the incentive scheme in more detail, with a focus on the communication interaction between opportunistic network nodes and the correlated security issues. User aliases in the form of a public key play a major role in accomplishing *authentication*, *non-repudiation*, and *integrity* of data and user behavior, as well as a means to preserve the privacy of a user. In addition, the Mediator, as introduced on page 51, is described in more detail.

Recall the basic idea: Participants are rewarded in the form of bonus points issued by the Information Producer. The dissemination process is based solely on the dissemination mechanisms described in Chapter 3. Since the mechanisms are based on individual user profiles, special care is taken to preserve user privacy. The incentive scheme allows only legitimate users to claim gained bonus points, while, at the same time, staying anonymous within the scheme. The way gained bonus points are used by a user afterwards is beyond the scope of the proposed scheme. We suggest one simple solution and point to the literature for more sophisticated schemes based on digital anonymous payment systems.

5.2.1 Roles

The incentive scheme distinguished between the following roles for opportunistic network nodes.

Definition 13 (Information Producer) An opportunistic network node is called *Information Producer*, if it is the source of an information item and has vital interest in disseminating that information item to a, a priori, unknown number of *Information Consumers*. □

The adPASS (see Section 6.1.1) prototype uses one to several stationary Information Sprinklers (recall Definition 4 on page 36). that are located at a shopping mall to act as Information Producers by disseminating digital advertisements from a co-located shop. Then, mobile nodes are able to learn about these advertisements, given that they are interested in the advertisements and within communication range.

Definition 14 (Information Consumer) An opportunistic network node is called *Information Consumer*, if it is the sink of an information item that was issued by an Information Producer. The Information Consumer uses the information item for a personal benefit. □

Taking adPASS as an example, a node acting as an Information Consumer uses the received digital advertisement to make a purchase.

Definition 15 (Information Bearer) An opportunistic network node is called *Information Bearer*, if it helps transport an information item from an Information Producer to an Information Consumer. The help can be rewarded by the Information Producer in the form of bonus points. □

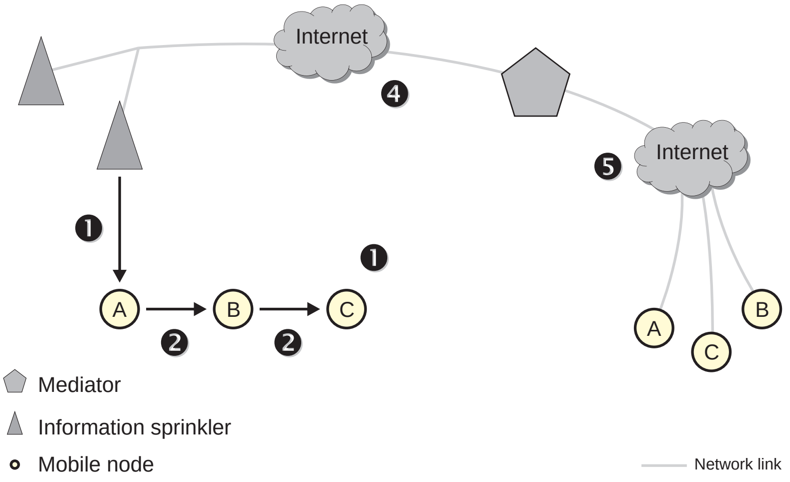


Figure 5.4: Incentive scheme communication pattern

It is possible that a mobile node acts as an Information Bearer and an Information Consumer at the same time. This means that the node uses the received information for its own benefit and passes the information further to other mobile nodes.

5.2.2 Mediator

The opportunistic network model, as described in Section 3.2, is extended by a central component called *Mediator* (see Section 3.6.2). The Mediator acts as a trusted third party between Information Bearers and an Information Producer. In this sense, *trusted* means that the Mediator and Information Producer do not collaborate by exchanging information that would support user profile creation. The Mediator's purpose is twofold:

- It acts as a central repository and keeps track of gained bonus points from participating nodes.
- It serves as an anonymizing proxy between an Information Producer and Information Bearers.

An Information Producer informs the Mediator about the user(s) who should be rewarded by bonus points. An Information Bearer queries the Mediator about the amount of bonus points he gained while participating in the incentive scheme.

5.2.3 Communication Pattern

Figure 5.4 displays an exemplary communication pattern of the proposed incentive scheme. It shows two Information Sprinklers that act as Information Producers.

They are connected to a Mediator, for example, using an Internet connection. Furthermore, there are three mobile nodes *A*, *B*, and *C*. *A* and *B* act as Information Bearers, while *C* acts as an Information Consumer. The distinct communication steps, labeled with numbers, are:

1. A mobile node *A*, while being in communication range to an Information Sprinkler, learns about an information item π. As a prerequisite, π must successfully match against *A*'s iWish-list.

2. Node *A* passes the information item on to *B* and user *B* itself on to *C*.

3. Node *C* takes some action upon receiving information item π. This step includes a bearer chain submission (see below) to the Information Provider.

   In the adPASS application, node *C* would visit a shop and buy the advertised product and tell the shop owner that the advertisement was received through user *A* and *B*.

4. From the bearer chain, bonus point information is extracted and submitted to the Mediator.

5. Information Bearers *A* and *B* query the Mediator about gained bonus points using a standard Internet connection.

The illustration in Figure 5.4 simplifies the setup in two ways. Firstly, the Information Sprinklers would be connected to a central hub, for example, a central server. This server would provide the connection to the Mediator as well as manage the information items for dissemination via the Sprinklers. Secondly, given that opportunistic network nodes consist of small mobile devices, for example, a 802.11b WiFi enabled PDA, an Internet connection to the Mediator might be established via a user's personal desktop computer that the mobile device is attached to.

5.2.4 Bonus Point Model

An Information Producer rewards Information Bearers in the form of bonus points. For this, an Information Producer assigns a maximum number of bonus points to dispense in case an information item reaches an Information Consumer. An intermediate node, i.e., an Information Bearer, is allowed to claim a certain share of *virtual* bonus points. These virtual bonus points will only become *real* bonus points if the information item reaches an Information Consumer and the consumer takes some action that is beneficial for himself and the Information Producer. In the adPASS application, this would be the purchase of an advertised product.

Information about preceding nodes is always transported along with the information item and stored in a *bearer chain*. This leads us to a graph theoretic model that is presented next.

5.2.5 Graph Theoretic Model

For the sake of simplicity, the model is restricted to the case of a particular information item π. The general case of multiple Information Producers and information items can be derived from this in a straightforward manner. Relations of Information Bearers passing along an information item π are modeled by a directed and weighted simple graph $G = G_\pi = (V, E, b)$.

The set of vertices $V = \{m\} \cup C$ consists of one Information Producer m and a set of Information Bearers $C \subseteq \{c_j : j \in \mathbb{N}\}$. By $E \subseteq V \times C$, we denote the set of edges of G. The mapping $b : E \to \mathbb{N}$ assigns a non-negative weight (number of bonus points) to every edge of G. To describe the timewise behavior of the system, we use a mapping $t : E \to \mathbb{R}$ whose values are interpreted as points in time.

For $(v, w) \in E$, we write $b(v, w)$ and $t(v, w)$ in the following as a shorthand for $b((v, w))$ and $t((v, w))$ respectively.

The Information Producer m is disseminating the information item for a certain period of time starting at the moment $t_0 \in \mathbb{R}$ using an Information Sprinkler.

The interpretation of an edge $(v, w) \in E$ is the following: an information item π was passed along from v to w at the moment $t(v, w) \geq t_0$. At this time, the intersection of the communication horizons of v and w was non-empty and, in addition, π matched an entry in w's iWish-list.

Because customers can only pass along information items they already have, this imposes a restriction for t, namely

$$\begin{aligned} (v, w) \in E \cap C^2 \Rightarrow \exists\, (u, v) \in E \\ t(u, v) < t(v, w). \end{aligned} \tag{5.2}$$

A *bearer chain* of length k from m to $c_{i_{k-1}}$ is described as a sequence $[e_j]_{j=0}^{k-1} = [e_0, \ldots, e_{k-1}]$ of edges $e_0 = (m, c_{i_0})$ and $e_j = (c_{i_{j-1}}, c_{i_j}) \in E$ for $1 \leq j \leq k-1$.

In the beginning, the Information Producer fixes the total number of (virtual) bonus points $b_0 \in \mathbb{N}$ which he will dispense for each successful dissemination to an Information Consumer. These points are shared among the participants in the chain $[e_j]_{j=0}^{k-1}$.

We let each Information Bearer c in the chain decide how many of the remaining bonus points to keep. This parameter influences the probability of an information item being passed along over a long distance. For every $c' \in C$ where $(c, c') \in E$, the value $b(c, c') > 0$ denotes how many bonus points c passes along to c'. A natural restriction on b is that c can only pass along less virtual points than he obtained before. Therefore, condition (5.2) is modified as follows:

$$\begin{aligned} (v, w) \in E \cap C^2 \Rightarrow \exists\, (u, v) \in E \\ t(u, v) < t(v, w) \wedge b(u, v) > b(v, w) \end{aligned} \tag{5.3}$$

Assume the last participant in the chain ($c_{i_{k-1}}$) decides to act on receiving the information item, for example to buy the advertised product at m's store. Then he gets the product at the price quoted in π and moreover the remaining bonus points

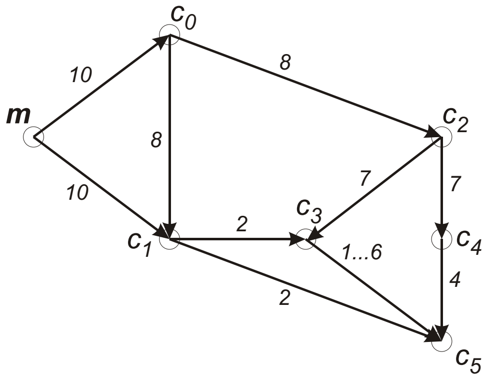


Figure 5.5: Example of bonus point passing

$b(e_{k-1})$. The other participants in the chain are granted the same amount of real bonus points as they kept virtual points when passing along the information item. These are $b(e_j) - b(e_{j+1})$ points for information bearer $c_{i_j}, 0 \le j \le k-2$.

adPASS Example: The example illustrates the incentive scheme on the basis of adPASS. We consider a graph with 6 participants, i.e. $V = \{m, c_0, \ldots, c_5\}$ and assume that c_5 will buy the product if he learns from the ad. If c_5 receive the ad via different communication chains, it will use the one with the higher amount of bonus points left. Merchant m assigns $b_0 = 10$ bonus points to the product. Figure 5.5 shows how the ad is then passed along.

- The bearer chain $[(m, c_0), (c_0, c_2), (c_2, c_4), (c_4, c_5)]$ expresses the following: Customer c_0 claims 2 points and passes $b(c_0, c_2) = 10 - 2 = 8$ points along. c_2 keeps a single point, i.e. $b(c_2, c_4) = b(c_2, c_3) = 7$ and finally c_4 keeps 3 points.

- If c_5 buys the product, then c_0, c_2, and c_4 are rewarded bonus points for successfully passing along the ad, while buyer c_5 gets the remaining 4 points.

- If c_0 were too greedy and claimed 8 virtual points, the ad would have been declined from c_4 in the above communication scenario because only one point remained. Therefore, the total number of bonus points is an upper bound for the number of hops.

- There is the possibility to learn from an ad via two different communication chains: $[(m, c_1), (c_1, c_5)]$, for instance, is another chain which transports the ad to c_5. However, the number of bonus points for c_5 is less ($b(c_1, c_5) = 2$).

- The system is time-dependent: Possible values for $b(c_3, c_5)$ are $1, \ldots 6$, depending on the relation of $t(c_1, c_3), t(c_2, c_3)$, and $t(c_3, c_5)$.

5.2.6 Extensions and Variants

It might be interesting to model and study different strategies of keeping and passing along bonus points. Without going into details, a short discussion on some variants and features of the model follows.

The example implicitly assumes that $b(c, c')$ is constant for a fixed c and arbitrary c' such that $(c, c') \in E$. While this appears to be a natural restriction, participants are allowed to vary the number of points they pass along. It is also assumed that $b(m, c) = b_0$ for all $c \in C$ such that $(m, c) \in E$. But the merchant may choose different values, e.g., for special offers limited in time.

Each participant c_{i_t} in the bearer chain $[e_j]_{j=0}^{k-1}$ may define a lower bound for one or more of the values $b(e_t), b(e_t)/b_0$ or even $(b(e_{t-1}) - b(e_t))/b_0$. This is a possibility to express a personal strategy or notion of fairness.

An approach which simplifies the system from the users' point of view is the following: In case a product is bought at the end of a bearer chain $[(m, c_{i_0}), \ldots, (c_{i_{k-2}}, c_{i_{k-1}})]$, the second-to-last (i.e. $c_{i_{k-2}}$) is rewarded the full bonus b_0 while the others get no bonus at all. This relieves the system of storing the values $b(e_j)$.

5.2.7 Security Goals

This section describes a number of typical security goals in general terms and discusses their particular meanings for the incentive scheme. Relevant security goals are briefly explained in the following:

- **Integrity**: Assure that information cannot be modified in any way without being detected.
- **Authentication**: Ensure the originality of some information (data authentication) or verify the identity of a party (entity authentication).
- **Non-Repudiation**: Prevent parties from denying having taken some action.
- **Anonymity**: Ensure that an entity remains unidentifiable within a set of parties.

This enumeration does not list all typical security goals. Goals such as *availability* or *authorization* are less important for the incentive scheme and therefore not considered.

Now, we will look into each security goal in turn and describe their meaning for participating parties and used entities.

Integrity First, the information item itself, for example, price information about an advertised product, should be kept safe from manipulation (information item integrity). Second, the integrity of the bearer chain that holds information about bonus point claims, needs to be ensured (bearer chain integrity).

Authentication For all opportunistic network nodes taking part in the incentive scheme, *authentication* of an information item provides the nodes with assurance that the information item was issued by the claimed Information Producer and not forged. This prevents nodes from sharing their personal resources in vain, since the Information Producer commits himself to issue a reward.

Non-Repudiation This goal prevents an Information Producer from denying that he has issued a certain information item, that might contain a certain offer.

Anonymity Information Bearers should be able to take part in the incentive scheme without revealing their identities to each other. This would hold off an attacker to create user profiles of personal preferences and thus protects user privacy. Also, an Information Producer should not be able to learn the identities of Information Bearers by analyzing the bearer chain.

5.2.8 Solutions

Having identified relevant security goals in the last section, this section describes appropriate solutions. The solutions combine cryptographic primitives, technical measures, and legal practice. Again, we look into each security goal in turn:

Authentication, Non-Repudiation, Information Item Integrity, and Anonymity The use of a public key infrastructure, for example provided by the Mediator, allows network nodes to authenticate information items. For this, an Information Producer m is required to use a key pair (P_m^-, P_m^+) that is certified by a certification authority (CA) under a certain policy. The certificate $Cert_{P_m^+}$ issued for the public key P_m^+ is called a *qualified* certificate. Now, during the initial dissemination of an information item π, an Information Producer binds his certificate to π and signs both:

$$S_{P_m^-}(\pi, Cert_{P_m^+}) := P \tag{5.4}$$

From now on, this signed tuple is called payload P. Whenever a node receives this payload, it is able to check:

(1) that the information item π was issued by m,

(2) that the information item π was not modified during the dissemination process. For example, the price of an advertised product has not been altered.

Thus, (1) and (2) hold as long as the signature verification succeeds. It follows from (1) that authentication and non-repudiation are achieved and (2) implies information item integrity.

If an Information Producer repudiates the dissemination of an information item, concerned Information Bearer and Information Consumer need to take legal actions. Here, the usage of qualified certificates will help them to prove their claims.

Bearer Chain Integrity To prevent a malicious node from manipulating the bearer chain, a chain from node an Information Provider m to an Information Consumer c_{i_k}, and nodes c_{i_0} to $c_{i_{k-1}}$ being Information Bearers, is secured the following way

$$\begin{aligned}
[\quad & S_{P^-_m}((P^+_m, P^+_{c_{i_0}}, b_{m,c_{i_0}})),\\
& S_{P^-_{c_{i_0}}}((P^+_{c_{i_0}}, P^+_{c_{i_1}}, b_{c_{i_0},c_{i_1}})),\\
& S_{P^-_{c_{i_1}}}((P^+_{c_{i_1}}, P^+_{c_{i_2}}, b_{c_{i_1},c_{i_2}})),\\
& \ldots\\
& S_{P^-_{c_{i_{k-1}}}}((P^+_{c_{i_{k-1}}}, P^+_{c_{i_k}}, b_{c_{i_{k-1}},c_{i_k}})) \quad]
\end{aligned} \tag{5.5}$$

Here $b_{c_{i_l},c_{i_m}} := b(c_{i_l}, c_{i_m})$ denotes how many bonus points c_{i_0} passes to c_{i_1} (see page 84). The general structure for one entry in the bearer chain is

$$(S, R, b)_{\texttt{signed by } S} \tag{5.6}$$

where S is an alias for the sender, R is an alias for the receiver and b is the number of bonus points passed from S to R. Both sender and receiver use public keys as aliases. While the first sender, i.e., the Information Producer, uses the public key certified in the payload P (see equation 5.4), all other participants in the chain use the public key out of their key bag (see page 79) of self generated keys as an alias. This method enables a participant to stay anonymous within the bearer chain and, at the same time, allows a participant to rightfully claim bonus points later on. Possession of the corresponding private key is proved without revealing the private key itself.

Bearer Chain Example: Using the adPASS Example from page 85, the bearer chain looks like the following

$$\begin{aligned}
[\quad & S_{P^-_m}((P^+_m, P^+_{c_0}, 10)),\ S_{P^-_{c_0}}((P^+_{c_0}, P^+_{c_2}, 8)),\\
& S_{P^-_{c_2}}((P^+_{c_2}, P^+_{c_4}, 7)),\ S_{P^-_{c_4}}((P^+_{c_4}, P^+_{c_5}, 4))] \quad]
\end{aligned}$$

Let us assume for the moment, that node c_5 is malicious and wants to manipulate the chain, for example, cut node c_0 and c_2 out of the bearer chain and claim more bonus points for himself. The new chain would have to look like this:

$$\begin{aligned}
[\quad & S_{P^-_m}((P^+_m, P^+_{c_4}, 10)), & (*)\\
& S_{P^-_{c_4}}((P^+_{c_4}, P^+_{c_5}, 4))] \quad]
\end{aligned}$$

But, to manipulate the first entry (*) that states the information was given from m to c_4, node c_4 has to know the private key of m to carry out the signing operation. This is not possible.

Table 5.1 summarizes applied techniques to reach the desired security goals. Since most of the techniques are based on public key cryptography, which in turn demand sufficient CPU power, a number of runtime tests were carried out during a prototype evaluation (see Chapter 6 for results).

Goal	Technique
Integrity	Digital signature operation
Authentication	Certificates
Non-Repudiation	Qualified signatures and certificates
Anonymity	Multiple key pairs as aliases

Table 5.1: Summary of protection goals and techniques

5.2.9 Bearer Chain Submission to the Mediator

Step 4 in Figure 5.4 illustrates bonus point information submission from an Information Producer to the Mediator. Given a bearer chain as defined in equation (5.5), the following list of tuples is sent to the Mediator.

$$\begin{aligned} [\ \ & (P^+_{c_{i_0}}, b_{m,c_{i_0}} - b_{c_{i_0},c_{i_1}})\,, \\ & (P^+_{c_{i_1}}, b_{c_{i_0},c_{i_1}} - b_{c_{i_1},c_{i_2}})\,, \\ & \ldots \\ & (P^+_{c_{i_k}}, b_{c_{i_{k-1}},c_{i_k}})\ \ ] \end{aligned} \tag{5.7}$$

Information Bearers are labeled $c_{i_0}, c_{i_1}, ..., c_{i_{k-1}}$ and the Information Consumer is labeled c_{i_k}.

Submission Example: Taking the adPASS example from page 85 again, the following list of tuple is submitted from the Information Producer to the Mediator.

$$[\ \ (P^+_{c_0}, 2)\,,\ (P^+_{c_2}, 1)\,,\ (P^+_{c_4}, 3)\,,\ (P^+_{c_5}, 4)\ \ ]$$

With this information the Mediator is able to reward all information bearers, who are able to prove legitimate bonus point claims (see next section).

5.2.10 Bonus Point Query

In the last communication step in Figure 5.4 (Step 5), an Information Bearer queries the Mediator about his participation in the incentive scheme. In other words, he wants to know, if his opportunistic network device contributed to a successful bearer chain. A chain is successful, if the information item reached an Information Consumer.

Since a query is transported over the Internet, information bearer privacy might be at risk, if the Mediator logs the bearer's IP address. One can circumvent this by using an anonymizing proxy or mix network [DM04, JAP00] that conceals the sender's IP-address.

To query the Mediator, we describe three schemes. Each scheme differs from the others with respect to privacy preservation and efficiency.

(1) The Mediator publishes a list of public keys that were part of a successful bearer chain, for example on a web page. An Information Bearer searches this list for the ones he holds the corresponding private key for. For each matching pair (P_i^+, P_i^-), the Information Bearer sends, in an individual session, the following tuple.

$$(S_{P_i^-}(P_i^+)) \tag{5.8}$$

In doing this, the Information Bearer proves that he is in possession of the corresponding private key and the Mediator issues the bonus points assigned to P_i^+.

This scheme offers the highest privacy protection. Using mix networks and multiple sessions, the Mediator cannot map a set of public keys to an individual bearer. Regarding efficiency, establishing individual sessions increases the communication overhead. In addition, searching the whole list of gathered public keys might put a heavy burden on the information bearer's device.

(2) For all key pairs $(P_{c_{i_j}}^+, P_{c_{i_j}}^-)$ out of the user's key bag KB, $\mid KB \mid= n$, (see Section 5.1) the information bearer sends the following list of tuples (in one session)

$$[\ (S_{P_{c_{i_1}}^-}(P_{c_{i_1}}^+)), (S_{P_{c_{i_2}}^-}(P_{c_{i_2}}^+)), \ldots, (S_{P_{c_{i_n}}^-}(P_{c_{i_n}}^+))\ ] \tag{5.9}$$

Again, each public key is signed by the corresponding private key and enables the Mediator to reward bonus points.

This scheme is more efficient. The bearer does not need to search a public key list and all communication happens in one session. On the other hand, all of the user's public keys are revealed together. This might lower the user's privacy if an attacker is able to somehow link different actions to a users's identity.

(3) For each key pair $(P_{c_{i_j}}^+, P_{c_{i_j}}^-) \in KB$ an Information Bearer sends in individual sessions, the following tuple.

$$(S_{P_{c_{i_j}}^-}(P_{c_{i_j}}^+)) \tag{5.10}$$

Secured by a mix network, this approach is most preferable. First of all, the burden of searching the list of all used public keys is put to the Mediator. Furthermore, the usage of individual sessions protects user's privacy, since profile creation is more difficult.

Assuming that the Mediator is a *trusted* third party, as introduced on page 82, all schemes offer sufficient privacy protection. Scheme (1) and (2) provide additional protection against a malicious Mediator who betrays users. Scheme (2) was implemented in the adPASS prototype (see Section 6.1.1).

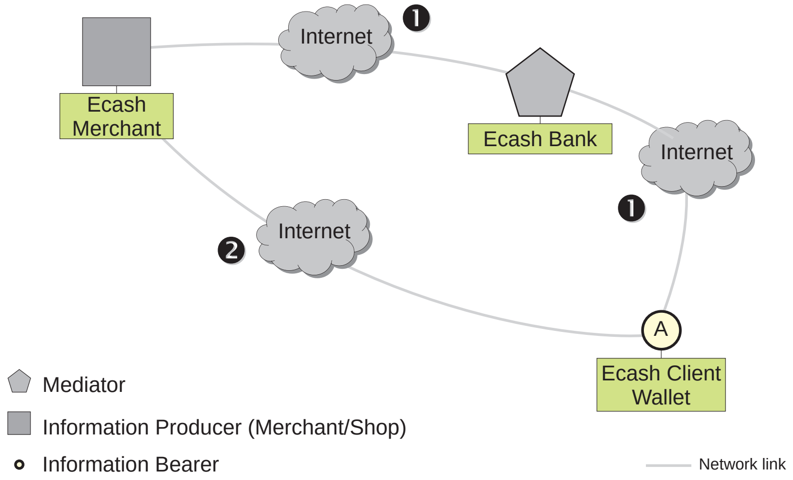


Figure 5.6: Incentive scheme communication pattern - Ecash extension

5.2.11 Bonus Point Payout

The proposed incentive scheme aims to increase user acceptability by rewarding participants in the form of bonus points. One key feature is the support for privacy preservation in every aspect of the scheme. Due to the use of self-generated key pairs and the avoidance of static data in the network layers, we allow users to rightfully claim gained bonus points without revealing their identities, i.e., users act anonymously.

At this point, the question arises of how a participant makes use of gained bonus points in some beneficial, personal way without harming his privacy and staying anonymous. As this is out of the scope of this thesis, we briefly outline how our incentive scheme components can be extended to support Ecash, one of the most prominent electronic cash payment systems, that allows fully anonymous secure payments on the Internet. Moreover, we describe a simple solution that is based on modified cash machines.

Anonymous Digital Payment Systems Extension Ecash is based on the work of David Chaum [Cha83, Cha85, CFN88] and provides the privacy of paper cash with the added security required for open networks like the Internet. The system allows clients to withdraw unique digital coins from an ordinary bank account in such a way that the bank does not learn the serial numbers of those coins. Thus, Ecash is fully anonymous. The Ecash system uses so-called *blind signatures* [Cha83]. It is described in detail in [OPT97, Sch97].

Figure 5.6 depicts the incentive scheme components defined in Section 5.2 with the added Ecash entities. The Mediator takes the additional rule of an Ecash bank.

Its task is to sign coins and to prevent the double-spending of coins. An Information Bearer owns an electronic wallet where withdrawn coins are stored and later used for electronic payment at an Information Producer. Thus, an Information Producer, e.g., a Merchant or shop, acts as an Ecash merchant and sells items by accepting payments from an electronic wallet. Figure 5.6 distinguishes three protocol steps within the payment:

1. After an Information Bearer has learned about his gained bonus points (see Section 5.2.10), he generates the same amount of electronic coins, each with a long random serial number. The coins are blinded by the Information Bearer and submitted to the Mediator for signing. The Mediator signs the coins, charges the Information Bearer's bank account and sends the coins back to him.

   The Information Bearer unblinds the coins and is now in the possession of electronic coins with a serial number that is unknown to the Mediator, in other words: anonymous digital cash.

2. In this step, the Information Bearer pays goods and services offered by the merchant, for example, a digital music download like Apple's iTunes [App03], using his coins. The coins are validated at the merchant (see next step) and the service is granted on success.

3. In this step the merchant validates the coins received from the Information Bearer to see if the coins have not been spent before. Thus, this step works hand-in-hand with step 2. In case the validation succeeded, the corresponding amount of money booked on the merchant's bank account.

Please note that the outlined steps only sketch the main ideas of combining Ecash with our proposed incentive scheme. Nonetheless, we showed that the incentive scheme components map easily to the Ecash entities. Thus, the Ecash protocols can be implemented in a straightforward manner.

Modified Cash Machines The simplest solution to keep users anonymous would make use of cash machines that are enhanced in two ways:

- The cash machine is connected to the Mediator, for example via the Internet.
- The cash machine offers a connection interface for the opportunistic network device. This may come in the form of a docking station or even a wireless link (Bluetooth or similar).

This setup would allow a user to query for bonus points at any of the typical wide-spread network of cash machines. Thus, bonus points would map to real money, for example, 1 EUR for 100 gained bonus points, and be dispensed right to the user. Since real money is not linked to any person, the user would stay anonymous. Needless to say, these cash machines are trusted in the same way the Mediator is trusted, i.e., they do not link or log public keys.

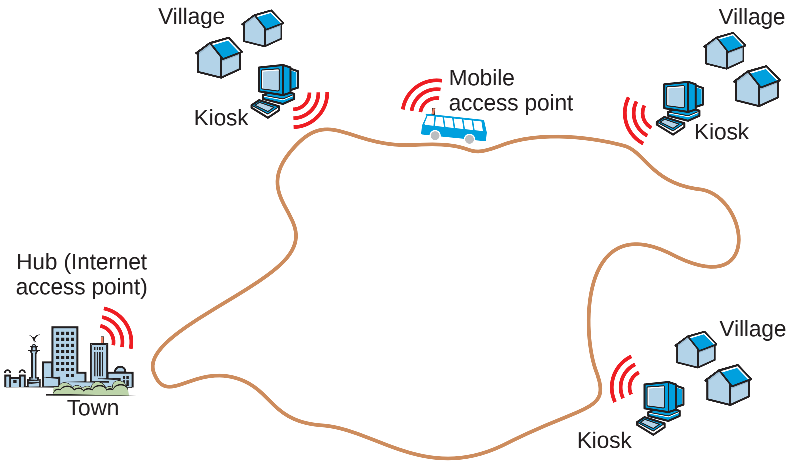


Figure 5.7: DakNet concepts

5.2.12 Application Example: adPASS

Within this thesis, the incentive scheme was implemented in adPASS. adPASS is an opportunistic network application that disseminates digital advertisements among interested users. Advertisements are issued by a vendor (Information Producer) that operates one or several Information Sprinklers, for example, within a shopping mall, to reach interested customers (Information Bearers/Consumers).

Since adPASS already served as a motivation for the incentive scheme in Chapter 3, we will not go into details here. An elaborated description will be given in Section 6.1.1.

5.2.13 Application Example: DakNet

The proposed incentive scheme is suitable for networks similar to opportunistic networks as we will see now. We use DakNet [PFH04, Sta05], a network with comparable opportunistic communication properties, as an example to sketch the applicability of the proposed incentive scheme. Implementation details are deliberately omitted.

DakNet System Description DakNet realizes a wireless ad-hoc network and provides asynchronous digital connectivity for developing rural areas. Asynchronous communication is based on a *store-and-forward* paradigm. So-called *mobile access points* (MAPs) exchange, store and forward information (email, voice mail, etc.) whenever they come into communication range. Bi-directional communication is possible, since the MAPs travel (physically) on fixed routes.

Figure 5.7 illustrates the DakNet concept (adapted from [PFH04]). Villages are connected to a town via MAPs that are installed on public buses. These buses operate on fixed routes several times a day. Making use of standard 802.11 WiFi technology, DakNet allows messages to be sent between so called *Information Kiosks*.

Let user *A* be located in the town and user *B* be located in a village. User *A* sends user *B* a message by syncing the message onto the Hub (see Figure 5.7). From the Hub the message is synced onto a MAP (public bus). The MAP physically transports the message close to the Information Kiosk in the village and syncs the message again. User *B* receives the message by accessing the Information Kiosk. User *B*'s reply is transported on the same way back to user *A*, it moves from an Information Kiosk to a MAP and from a MAP further to the Hub (located in town).

DakNet Incentive Scheme Extension The DakNet *store-and-forward* communication paradigm is very similar to the *information move* mechanism (see Section 3.4.3) in opportunistic networks. Due to this similarity, the incentive scheme maps well to the DakNet system setup. We describe this mapping now:

The Internet Service Provider (ISP), located in town, takes on the role of the Information Producer. He issues bonus points for message delivery to users located in remote villages. Thus, a MAP takes the role of an Information Bearer. There may be several MAPs, for example, buses from different companies, that compete in delivering messages. Finally, a user who is located in a town takes on the role of an Information Consumer. In detail, a message delivery from a user *A*, who lives in a town, to a user *B*, who lives in a village, is as follows:

(1) User *A* creates a message and submits the message for delivery to the Information Producer (ISP). The producer assigns a number of bonus points to the message.

(2) One or several MAPs, in the role of Information Bearers, download the message from the Information Producer, reserve a share of bonus points and physically transport the message (in part or in total) to the village's kiosk.

(3) User *B* downloads the message from the kiosk.

Each step includes the creation/extension of a bearer chain. Thus, user *B* knows the public keys of MAPs that helped transport the message to him. This information is extracted and sent back to the Information Producer (using the same mechanism). Next, the Information Producer submits the bearer chain bonus point claims to a Mediator, where MAPs can request them later.

Differences When applying the incentive scheme to DakNet, there is a major difference concerning user privacy and user identification. In order to work, DakNet asks for the unambiguous identification of sender and receiver. Thus, we implicitly assume some kind of public key infrastructure that assigns email addresses to key pairs. This also enables the sender to encrypt the message with the receiver's public key.

Next, since the MAPs just transport messages, their privacy is not at risk. On the contrary, the communication pattern between user *A* and user *B* is revealed to MAPs. This could be alleviated, if the Information Producer collects a number of sender messages, puts them all into one message and encrypts it with an Information Kiosk's public key. The Kiosk itself would decrypt it and forward the individual messages to the different receivers.

Proximity Based Services Similar to DakNet, Lueg and Mahmood [LM04a, LM04b] describe a system to update electronic bus schedules in rural and remote areas in Australia that lack network coverage. Their approach, called *mobile data recharging (MDR)*, enables mobile timetable recharging using the following components. A stationary *base station* located at the bus depot maintains the most recent version of the bus schedule. This schedule is downloaded onto buses upon departure. Upon arrival at a bus stop, the new timetable is uploaded at the bus stop's *recharge station*. All communication and data synchronization is done using a wireless link. The authors also point out that wireless access points integrated into bus stops could provide other local information like geographic coordinates or information regarding nearby attractions. This leads to our proximity based services as already described within this work in detail (see page 39).

5.3 Summary

This chapter addressed user acceptability in opportunistic networks in two ways. First, it presented means to preserve user privacy by avoiding static and therefore traceable data in the network stack. This approach is feasible due to the *one-hop* communication paradigm in opportunistic networks. Second, an incentive scheme based on bonus points was described. It allows participants to rightfully claim bonus points within a dissemination chain while staying anonymous.

Both privacy preservation and the incentive scheme, two important human aspects present in opportunistic networks, aim to increase the user acceptability of opportunistic network applications.

Chapter 6

Technical Feasibility

As part of this work, various aspects of opportunistic networks have been evaluated with a focus on the proposed data dissemination mechanism and the privacy preserving scheme. The effectiveness of the data dissemination protocol has been investigated by simulation (see next chapter).

This chapter presents evaluation results addressing various *technical* feasibility aspects of opportunistic networks. We have investigated the following questions:

1. Is current off-the-shelf hardware and software, especially small mobile devices, suitable for opportunistic network applications and especially for the data dissemination protocol?

2. Is the privacy preserving scheme feasible, i.e., does current off-the-shelf hardware provide sufficient computational power to implement the proposed scheme?

3. Is the proposed privacy preserving scheme adequate to protect a user's privacy needs?

In order to answer these questions, we have developed two prototype applications, adPASS (see Section 6.1.1) and musicClouds (see Section 6.1.2), running on Windows CE PDAs. These prototypes provide us with a testbed to carry out a number real world experiments. We conduced several runtime measurements to prove technical feasibility aspects of opportunistic networks.

The remainder of this chapter is organized as follows. Section 6.1 presents the prototypes with respect to technical realization and implemented functionality. Section 6.2 evaluates our data dissemination protocol by conducting real world experiments. The privacy preserving method is evaluated in terms of runtime measurements and appropriateness considerations in Section 6.3. We summarize this chapter in Section 6.4.

6.1 Prototypes

This section describes the prototypes that have been developed in order to carry out the real world experiments presented in Section 6.2. The prototypes implement our opportunistic network architecture described in Section 3.2.1.

6.1.1 adPASS

adPASS [SH04, HS03] is a system for spreading digital advertisements (ads) among interested users. Each user specifies his interests in a profile that is stored on the mobile device. The communication scheme resembles the way information is spread by word of mouth between human beings, e.g., when recommending something to someone else.

As an incentive for users to take part in the system, adPASS provides an anonymous bonus point model that rewards a user who carries an advertisement on the way from the vendor to a potential customer.

Ranganathan and Campbell [RC02] discuss mobile advertising in pervasive environments and outline several challenges. Here, adPASS contributes in two ways: First, it is an example for *serendipitous* advertising and second, adPASS provides means to deliver advertisements to the right people without harming their privacy.

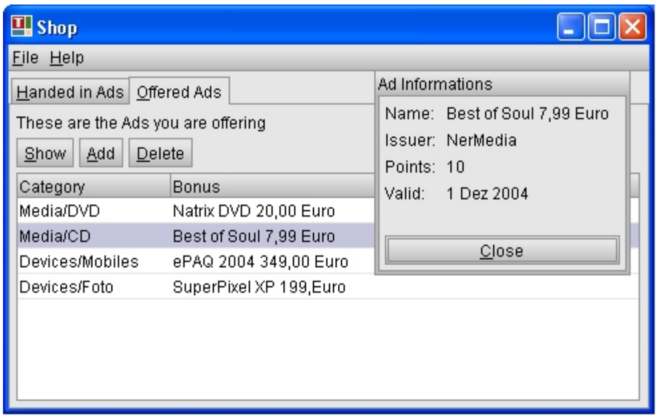
An introduction to adPASS was given in Chapter 3. The adPASS incentive scheme was described in detail in Section 5.2. This section provides some technical details on the adPASS prototype, its provided functionality from a user's point of view, as well as a typical usage scenario.

Technical Details The adPASS prototype is divided into several software packages that implement different roles, namely the Information Producer, Information Bearer and Information Consumer, as described in the last chapter (see Section 5.2.1), and the Mediator. All implementation was done in the Java programming language. The runtime environment for the Information Producer and Mediator is the standard Java2 Virtual Machine (VM) [Sun06] and for a Information Bearer/-Consumer a Java2 Micro Edition VM [Neg06] with its restricted capabilities. For all wireless communication, 802.11b WiFi is used in ad-hoc network mode.

For the concrete experiments, the Information Producer, acting as an Information Sprinkler, runs on a PC with Windows XP as the underlying operating system. The same holds for the Mediator. For the mobile nodes, iPAQ PDAs (Vendor Compaq, Model 3870) were used. These devices only offer limited resources: their Processor is an Intel SA-1110 (206 MHz) and the available memory is 64 MB (RAM). The iPAQ operates under Windows CE Version 3.0.

Functionality Figure 6.1 shows a screenshot of the adPASS producer, in our case a shop that offers several consumer media products (CDs, DVDs, Books,) and home entertainment hardware.

The producer application offers two main functionalities:



These are the Ads you are offering

Category	Bonus
Media/DVD	Natrix DVD 20,00 Euro
Media/CD	Best of Soul 7,99 Euro
Devices/Mobiles	ePAQ 2004 349,00 Euro
Devices/Foto	SuperPixel XP 199,Euro

Name: Best of Soul 7,99 Euro
Issuer: NerMedia
Points: 10
Valid: 1 Dez 2004

Figure 6.1: adPASS Information Producer (shop) screenshot

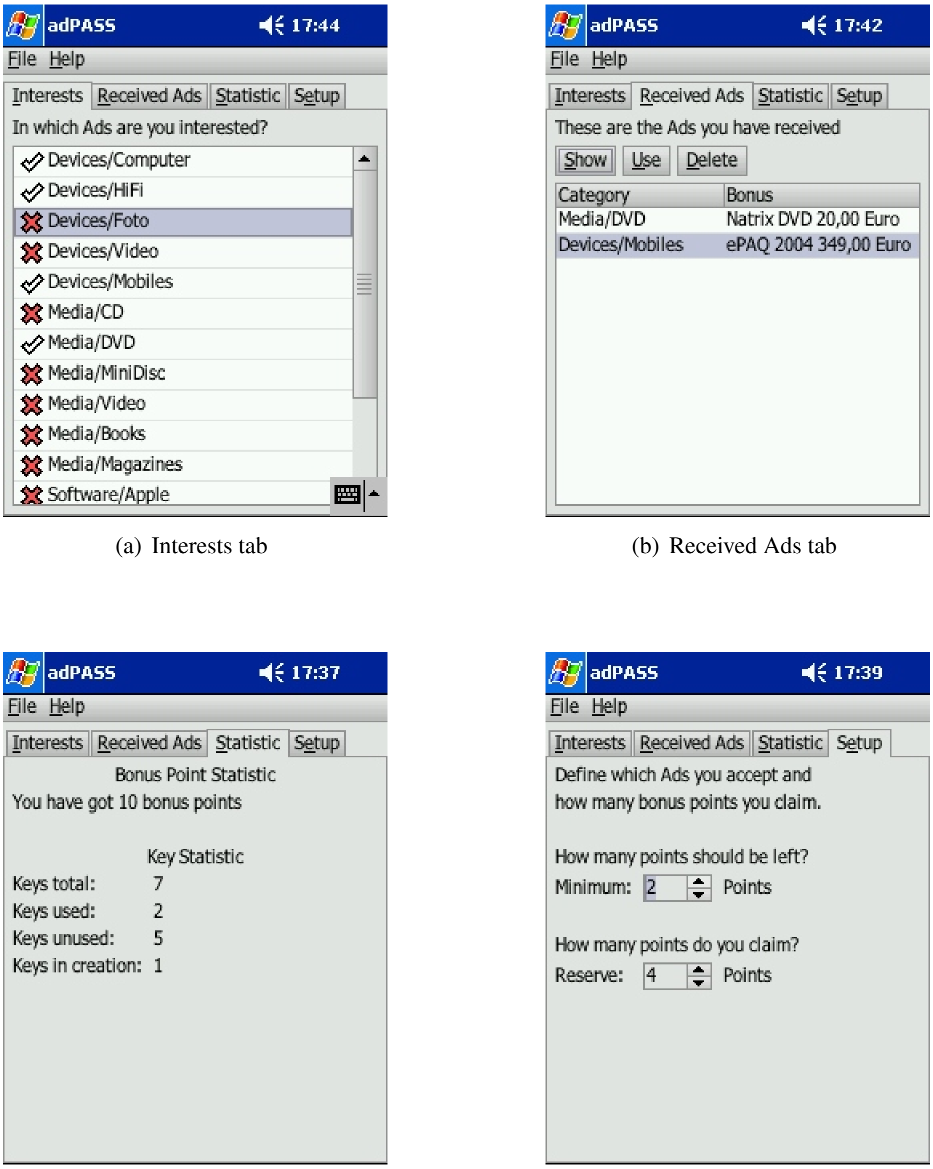
- Creation and deletion of advertisements (`Offered Ads` tab in Figure 6.1).
- Acceptance of handed in advertisements (`Handed in Ads` tab in Figure 6.1).

During advertisement creation, the producer sets the product name, associated product category, price, validity period, and the amount of bonus points he is willing to issue in total on a successful purchase of the advertised product.

A consumer's ad that is handed in during a purchase will show up in the `Handed in Ads` tab. If the Information Producer accepts the ad, i.e., the ad is valid and was issued by him, the bearer chain is submitted to the Mediator (see protocol details in Chapter 5.2.9) and bonus points can be redeemed later by Information Bearers.

Figure 6.2 shows screenshots of the adPASS mobile node. The application GUI is organized into four tabs. The first tab (Figure 6.2(a)) allows a user to express his interest in a certain product category. The node will collect (and pass along) all advertisements that belong to the selected categories. The second tab opens an overview of already received ads (Figure 6.2(b)). A user may view details of the ad to see whether the ad is useful. With the `Use`-button, a user hands in the ad at a shop. The `Statistic`-tab displays information about the amount of bonus points gained for later use and a report on the key generation and usage process. Recall from Chapter 5.1 that a user stays anonymous by generating a set of public keys as aliases. Our prototype always keeps a minimum of 6 key pairs at hand, which can be configured at program startup. Whenever the application is idle, the key generation process is triggered. With the fourth tab (Figure 6.2(d)), a user specifies the minimum amount of bonus points left in an advertisement to be of interest for him and the maximum amount of bonus points he is going to claim for himself (set to 4 points in the screenshot).

Typical Usage Scenario The ideas behind adPASS were already discussed on page 50. The roles and interaction pattern have been described in detail in Sec-



In which Ads are you interested?

- ✓ Devices/Computer
- ✓ Devices/HiFi
- ✗ Devices/Foto
- ✗ Devices/Video
- ✓ Devices/Mobiles
- ✗ Media/CD
- ✓ Media/DVD
- ✗ Media/MiniDisc
- ✗ Media/Video
- ✗ Media/Books
- ✗ Media/Magazines
- ✗ Software/Apple

(a) Interests tab

These are the Ads you have received

Category	Bonus
Media/DVD	Natrix DVD 20,00 Euro
Devices/Mobiles	ePAQ 2004 349,00 Euro

(b) Received Ads tab

Bonus Point Statistic

You have got 10 bonus points

Key Statistic

Keys total:	7
Keys used:	2
Keys unused:	5
Keys in creation:	1

(c) Statistic tab

Define which Ads you accept and how many bonus points you claim.

How many points should be left?

Minimum: 2 Points

How many points do you claim?

Reserve: 4 Points

(d) Setup tab

Figure 6.2: adPASS mobile node screenshots

tion 5.2. For the sake of readability, we will briefly recapitulate major points now. A typical usage scenario comprises the following steps:

1. An Information Producer (here an owner of a shop) generates advertisements for dissemination via adPASS. Information Sprinklers broadcast these advertisements within the vicinity of a shop.

2. A mobile node in communication range of an Information Sprinkler copies ads it is interested in on to the device. The node claims a certain amount of bonus points that come with the ad. The ad is then passed on to other users by serendipity encounters with other nodes. This leads to the construction of the bearer chain.

3. A user that makes use of the ad and buys the advertised product visits the shop. At the same time, the bearer chain is given to the shop and in turn passed on to the Mediator.

4. adPASS participants periodically query the Mediator for successful advertisement propagation that lead to a bonus point reward. These payouts are used later by the participants.

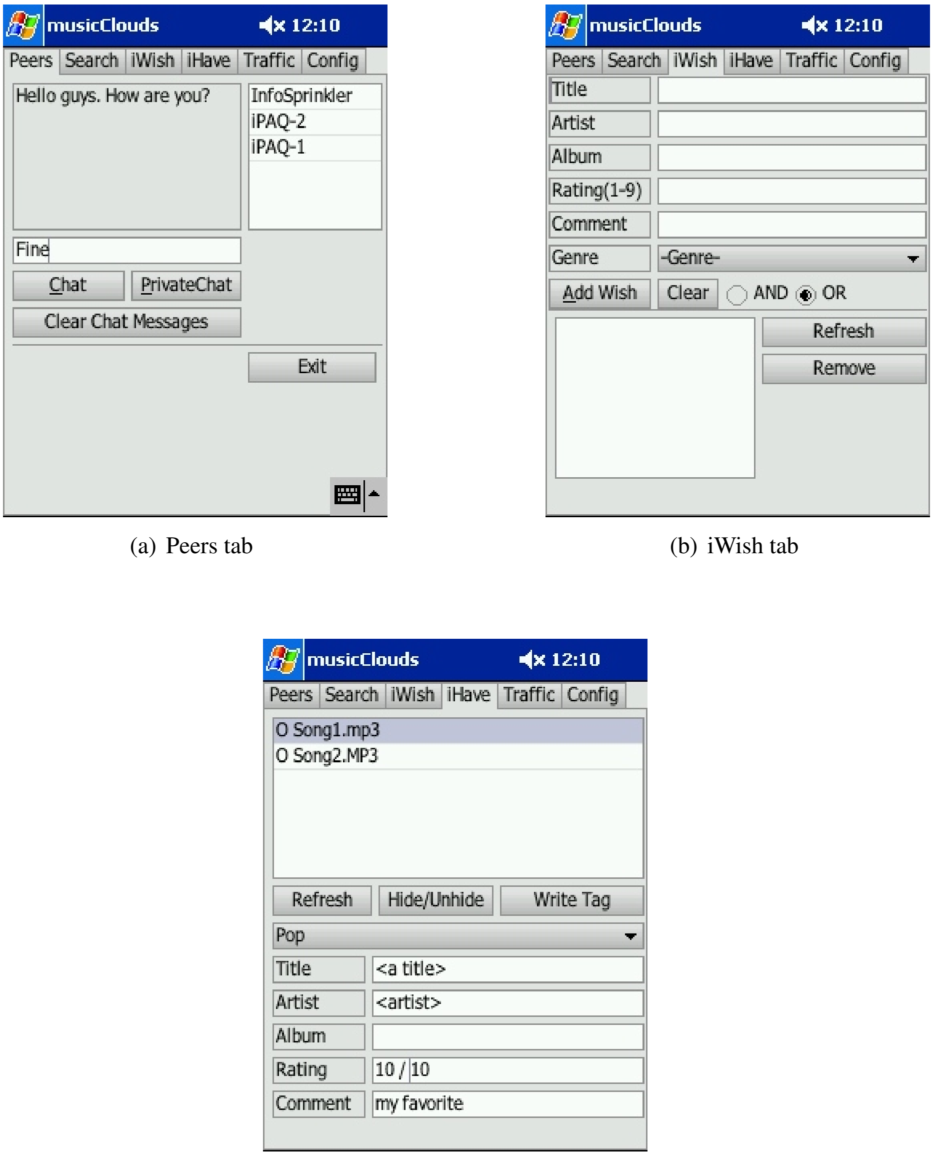
The adPASS prototype is used mainly to evaluate our proposed privacy preserving scheme (see Section 6.3), since the need for privacy preservation is crucial in opportunistic network applications of this type.

6.1.2 musicClouds

Our second prototype *musicClouds* allows opportunistic network nodes to share music files in an autonomous manner. For this, a user specifies search patterns beforehand.

Technical Details Again, the implementation was done in the Java programming language. musicClouds runs within the before-mentioned Java2 Micro Edition VM on iPAQ PDAs and 802.11b WiFi was used for wireless ad-hoc communication. Music files are encoded in mp3 format. For information tagging (see Section 3.5) the ID3 meta format was used [NM05].

Functionality Organized as tabs, the GUI offers the user several functions. Figure 6.3(a) displays active nodes in communication range and allows chat messages to be exchanged with all other users or just one prior selected user (`PrivateChat` button). Figure 6.3(b) depicts the `iWish` tab. Here a user inputs his search pattern. Fields map onto the corresponding ID3 attributes with the exception of `Rating`. There is no rating field in ID3. The implementation uses the first 2 bytes of the comment field to store a rating value. Thus, the comment space is slightly reduced. The individual entries are combined by *and* or *or* boolean operations (see radio



Hello guys. How are you?

InfoSprinkler
iPAQ-2
iPAQ-1

Fine

(a) Peers tab

Genre: -Genre-

○ AND ● OR

(b) iWish tab

O Song1.mp3
O Song2.MP3

Pop

Title: <a title>
Artist: <artist>
Album:
Rating: 10 / 10
Comment: my favorite

(c) iHave tab

Figure 6.3: musicClouds mobile node screenshots

button). The iHave tab shows the already received music files and their ID3 meta information (Figure 6.3(c)). It is possible to fill incomplete meta information and write an update onto the music file by pressing the Write Tag button. In addition, a user might want to exclude a certain file from sharing with others (Hide/Unhide button). There are two more tabs. The Traffic tab shows current file upload and download traffic. The current musicClouds node implementation allows only one upload and one download to happen in parallel. Although it would be possible to allow several uploads and downloads to happen in parallel, we believe this does not improve performance, since the wireless link is a shared (broadcast) medium. In addition, since each upload and download task asks for another program thread, preliminary tests have revealed during implementation that this may put too much burden on the virtual machine.

Finally, several program parameters can be adjusted via the Config tab, for example the timeout values for node discovery or the name of the node.

Typical Usage Scenario A typical setting for musicClouds is a concert or music festival. These events bring together people with a similar music taste. musicClouds can help to share music between participants during the event. In addition, the chat function will help people get into contact with each other and make new friends.

Another setting is more commercially driven. An Information Sprinkler offers music samples of the latest songs, for example in a music store or shopping mall. These samples, picked up by mobile nodes (using musicClouds) are spread among interested parties as a sample matches entries in the iWish tab.

The musicClouds prototype is used mainly to evaluate the performance of the proposed data dissemination protocol, i.e., to show the feasibility with off-the-self hardware in realistic setups. The results are presented in Section 6.2.

Having described the prototypes, including technical details and their use, the next section presents the real world experiments we have conducted using the prototypes and the results we have obtained.

6.2 Real World Experiments

We aim to evaluate the technical feasibility of opportunistic networks. Thus, we use our prototypes in various experiments to evaluate real world settings. For our experiments, we used the following hardware:

- One *Toshiba e740* PDA (Intel PXA250 400 MHz Processor, 64 MB RAM, 32 MB ROM, OS: Microsoft PocketPC 2002)
- Three *Compaq iPAQ 3870* PDA (Intel SA-1110 206 MHz Processor, 64 MB RAM, 32 MB ROM, OS: Microsoft PocketPC 2002)
- one *Compaq Armada 1700* notebook (Intel Pentium II 233 MHz, 192 MB RAM, 5 GB HD, OS: MS Windows 2000)

(a) Toshiba e740

(b) iPAQ 3870

Figure 6.4: Evaluation platforms running the iClouds prototype

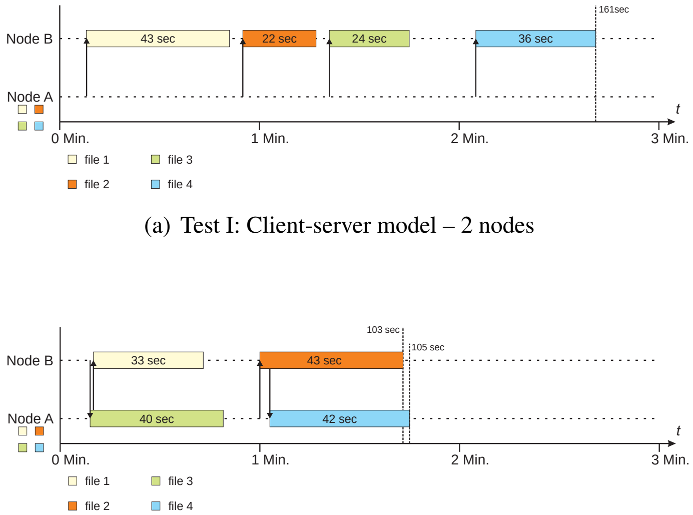
The PDAs are shown in Figure 6.4. All devices used 802.11b WiFi, either via internally or externally attached cards. The e740 PDA has an integrated 802.11b WiFi network interface. For the iPAQs we use an extension pack with standard PC-card based 802.11b WiFi network interfaces. Throughout all the following figures, node *A* denotes the Toshiba and nodes *B* – *D* denote the iPAQs. The notebook acts as an Information Sprinkler and is abbreviated as *IS*.

All experiments are based on the musicClouds prototype. For dissemination, we used four mp3 files of equal size (≈ 3.8 MB, playing time ≈ 4 minutes), but with different names and ID3 meta tag entries.

We carried out seven tests in total. Tests are labeled by Latin numeric characters I – VII. In tests I – V, nodes are immobile, while tests VI and VII take a pedestrian mobility behavior for nodes into account.

Test I: Client-server model – 2 nodes This tests consists of two nodes (PDAs) that are 10 meters away from each other in direct line of sight.

For node *A* we used the Toshiba PDA and for node *B* we used one of the iPAQs PDAs in order to have some heterogeneity between the nodes. Node *A* offers all four mp3 files and is interested in none, i.e., his iHave-list has four entries and his iWish-list is empty. Node *B* wants all of these files and has none, i.e., the iHave-list and iWish-list setup is just the opposite of node *A*. All in all, the communication resembles a client/server setup with node *A* being the server and node *B* being the client. Figure 6.5(a) depicts the observed behavior for test I. In total, it took 161 seconds to transfer all files (≈ 14,6 MB) from node *A* to node *B*. The net aggregate transmission time for all files is 125 seconds with a channel utilization of about 978 kBit/s and a mean transfer time per file of about 31 seconds. The observed channel utilization is significantly lower than the theoretical bandwidth of 11 MBit/s for the 802.11b WiFi cards used.



(a) Test I: Client-server model – 2 nodes

(b) Test II: Peer-to-Peer model – 2 nodes

Figure 6.5: Test I & II: musicClouds download behavior - 2 nodes

Test II: Peer-to-Peer model – 2 nodes The setup of this test is similar to test I. Different is the nodes' sharing behavior. Both nodes own two files and both nodes are searching for the other two missing files, i.e., node *A* owns file 1 and 2 and is looking for 3 and 4, while node *B* owns 3 and 4 and is looking for 1 and 2. The observed performance is depict in Figure 6.5(b). The download finishes after 103 seconds for node *B* and slightly later for node *A*. Now, we observe that the mean transfer time per file is about 25% higher (39 sec) with a 20.5% lower channel utilization that before.

Notably in both tests, the gaps between the download times vary a lot. They start from 4 seconds up to 20 seconds. The reason for this is found in the implementation. Parallel to the download process, each node periodically (every 20 seconds[1]) broadcasts its iWish-list. If a receiving device is busy, this means there is already a download process established with the sending device. Therefore, the newly received iWish-list is ignored. After a download finishes, it may take up to 20 seconds to receive a new iWish-list from a close-by node before a new download process is established.

Test III: Client-server model – 4 nodes This test consists of four nodes (PDAs). The distance between nodes varies between 8 – 10 meters with direct line of sight. In this test, node *A* owns all files at start-up. All other nodes (*B*, *C* and *D*) are interested in the files. The observed behavior (from a download perspective) is depicted in Figure 6.6. This test clearly reveals the data dissemination property of our protocol.

[1] Configurable at start-up time.

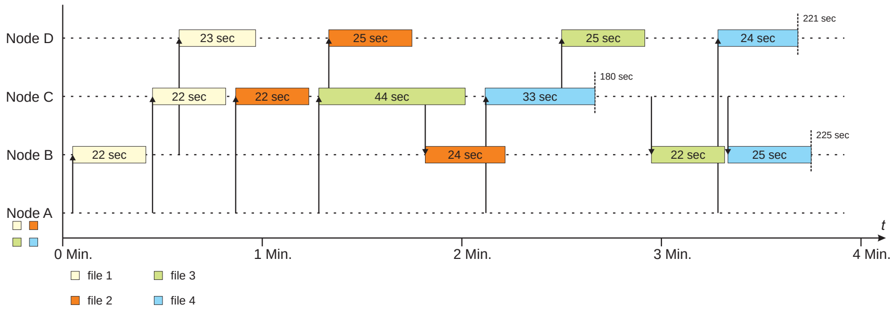


Figure 6.6: Test III: Client-server model – 4 nodes

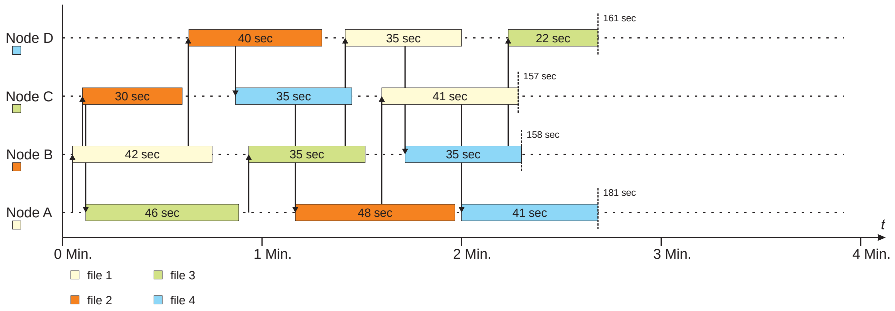
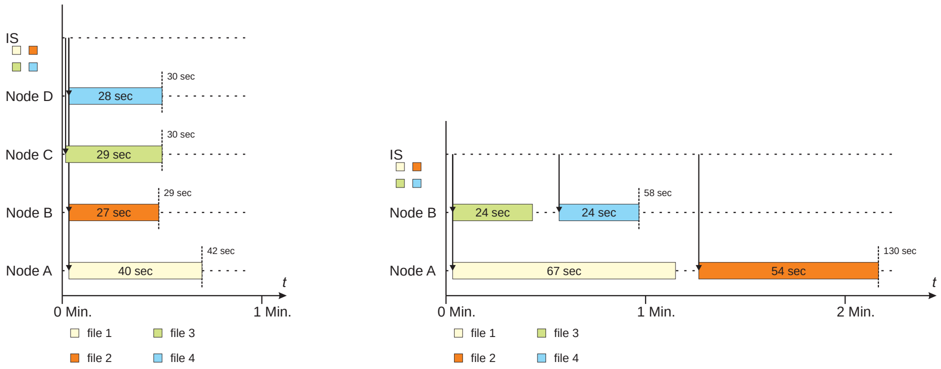


Figure 6.7: Test IV: Peer-to-Peer model – 4 nodes

For example, first node A shares file 1 (yellow box) with node B. Second, while node A shares file 1 with node C, node B shares the same file with node D, thus taking the server role this time. The same behavior can be observed later on, for example, node C shares file 3 with node D and B. The average download time per file is 26 seconds.

Test IV: Peer-to-Peer model – 4 nodes The test varies from test C in the initial data setup. Here, each node owns one file at the beginning and is interested in all three missing files. Figure 6.7 displays the system behavior from a download perspective. Again, several times we observe the multi-hop dissemination behavior. For example, file 3 is passed from node D to node C and from node C further to node A. The idle state of node D right at the beginning comes from the fact that each node carries out one upload and one download in parallel at a time. By chance, node A, B and C form a kind of circle, i.e., node A downloads from node C and node C downloads from node B, which itself downloads from node A. This blocks out node D. In this test, the average download time per file is 37.5 seconds.

Up to now, all tests used comparatively equal hardware, i.e., PDA class nodes, according to CPU power, amount of memory and battery power. The next two tests introduce one stronger node, i.e., the Compaq notebook (see technical details on



(a) Test V: 4 nodes, 1 IS – static nodes

(b) Test VI: 2 nodes, 1 IS – moving nodes

Figure 6.8: Test V & VI: Client-server model – Information Sprinkler tests

page 103), to act as an Information Sprinkler.

Test V: Client-server model – 4 nodes, 1 Information Sprinkler This setup consists of one Information Sprinkler (IS) that provides all four files and four mobile nodes. Each node is interested in one of the four files. Download results are shown in Figure 6.8(a). We observe that node *A*'s download behavior is about 1/3 slower compared to the other nodes. This follows from node *A*'s less powerful hardware (Toshiba e740, see page 103), especially the embedded and smaller antenna.

So far, the tests did not take node mobility into account. In the next two tests, mobile nodes move at pedestrian speed (≈ 1m/s).

Test VI: Client-server model – 2 nodes, 1 Information Sprinkler Again, the Information Sprinkler offers all four files. Node *A* is interested in file 1 and 2, whereas node *B* is interested in file 3 and 4. Within a radius of 20 meters, both nodes move randomly around the Information Sprinkler. The observed behavior is depicted in Figure 6.8(b). It takes approximately two times longer for node *A* to download both files in comparison to node *B*. Thus, a strong antenna is even more important for moving nodes. On the other hand, an overall download period of 130 seconds for two complete mp3 music files is sufficient to realize a proximity based service for advertising newly released songs, especially since an advertisement or teaser would only include a snippet of the song. For example, the online shop *amazon.de* advertises music CDs by providing 30 seconds song snippets for more than 100.000 CDs. As an example, a customer that stays 5 minutes within communication range of an Information Sprinkler could learn about 35 newly released songs if he uses a Toshiba e740 or even 171 newly release songs if he carries an iPAQ with him.

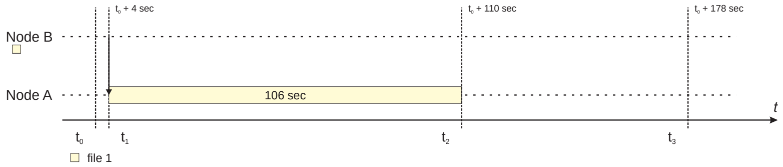


Figure 6.9: Test VII: Client-server model – 2 nodes, one passing by another

Test VII: Client-server model – 2 Nodes, one passing by another Whereas in all experiments described before the nodes are within communication range right from the beginning, i.e., they discover each other right away, in these experiments two nodes are activated while being out of communication range. Then, node *A* moves at about 1m/s speed in the direction of node *B*. Thus, the two nodes have to discover each other before a download may happen. In our case, node *B* offers a music file node *A* is interested in. Node *A* passes by node *B* at a distance of about 5 meters and leaves in the opposite direction. Imagine node *B* being in a car and waiting at a traffic light, with node *A* passing by on foot. The results of this experiment are shown in Figure 6.9. At time t_0, node *A* and *B* discover each other. 4 seconds later the download of file 1 begins. The download takes 106 seconds. After the download has finished, the two nodes see each other for another 68 seconds, before they lose contact. Thus, the observed overall time window to exchange files is 178 seconds. With a walking speed of 1 m/s, the theoretical time window is 199.74 seconds[2]. The observed time window is smaller due to inferences and signal absorption from the surroundings.

Discussion The mean application payload throughput for all experiments is about 955,2 kBit/s, which is significantly lower than the theoretical throughput of 11 MBit/s for the 802.11b WiFi network interface cards used. We assume that this is due to the interoperability requirement of 802.11b WiFi cards with older 802.11 WiFi cards as demanded in the standard specification [IEE99]. Although 802.11b WiFi cards may implement dynamic rate switching with the goal of improving performance, control frames like RTS, CTS and ACK must be transmitted at a lower rate that belongs to the so-called *basic rate set*. This allows older 802.11 WiFi cards to detect control frames and thus participate in a network. Anastasi et al. [ABCG03, ABCG04] carried out a couple of experiments to measure 802.11b WiFi performance. They measured TCP and UDP traffic at the application layer and observed similar throughput values to our perceived results. Another reason might be a broken network stack implementation on the mobile devices. This needs to be investigated further by making use of other network stack implementations, which we did not have on hand during our experiments.

[2] $x = 2 \cdot \sqrt{10000 - 25}$, with 100 meters communication range and 5 meters passing distance between node *A* and *B*.

Despite the observed reduction in throughput, our experiments clearly illustrate that the throughput provided by off-the-shelf hardware is already adequate for our data dissemination protocol. An open issue is the amount of energy consumption of 802.11b WiFi cards. Since opportunistic network applications need to run all the time, the PDA and network interface cards have to be active. We performed a long-term experiment, where one iPAQ node is requesting a file, deletes it upon receipt, and requesting it again and so on. We found that the battery lasts for approximately 2 hours and 41 minutes. This is clearly insufficient for practical purposes, but on-going improvements in fuel cell technology to increase online capacity of mobile phones [NTT05] lead the way.

Another issue we deliberately left aside in our considerations is connection loss or interruption due to bad radio reception, signal inferences or other obstacles. Whenever a connection link breaks before the information of interest is fully transmitted the already received part is discarded. Since opportunistic networks are based on spontaneous and unplanned user encounters, heavily influenced by user mobility, aborted transmissions do not violate the opportunistic network model. Aborted transmissions are treated as if they did not happen in the first place.

6.3 Privacy Preserving Method Evaluation

Our method for preserving user privacy is based on avoiding static data which can be linked to an individual user. On the application layer, this is done by using multiple self-generated key pairs. This is crucial for adPASS and similar applications that otherwise would disclose too much private information. Thus, in Section 6.3.1 we present a set of runtime measurements related to our key usage.

Afterwards, Section 6.3.2 argues why our proposed method is appropriate for opportunistic networks.

6.3.1 Runtime Measurements

In order to preserve a user's privacy within adPASS, a mobile node needs to carry out various cryptographic operations; in detail: key generation, signing, and validation. Taking the limited resources of a mobile node (comparatively slow CPU, less memory) into account, we conducted several runtime measurements in order to show the feasibility of this approach.

As signature scheme, we used standard RSA combined with the SHA-1 hash function. We measured the time for generating a key pair, signing and verifying for key lengths from 384 to 1536 bit. For our runtime tests we used the following components:

- iPAQ Pocket PC (Windows CE 3.0), Intel SA 1110 Proc. 206 MHz, 64 MB RAM
- CrEme Java VM [Neg06]

key	operation runtime (sec)		
bits	key gener.	signature	verification
384	6.28	0.10	0.03
512	12.67	0.18	0.04
640	28.76	0.31	0.06
768	44.79	0.48	0.07
1024	123.54	0.99	0.10
1280	226.45	1.69	0.12
1536	298.91	2.98	0.18

Table 6.1: Average timings for RSA (PAQ Pocket PC)

- Cryptix JCE Provider [Var06]

The timings in Table 6.1 are averaged over 20 executions of the respective operation.

Note that the task of creating a new key pair is in general very time-consuming on the PDA. This can be circumvented easily by using key pairs that were created on a desktop PC in advance and then copied to the mobile node's key bag (see page 79). This speeds up the computation by a factor of 20 to 40. The demand for memory capacity is negligible (about 1-2KB per key pair for the key sizes in question). Also, key pairs can be deleted on the mobile node when changing the identity, since it is possible to back them up on the PC.

Only the operation of signing the bearer chain must be accomplished in real time during the protocol (see Section 5.2.8) while two nodes are in communication range. Our experiments show that this is easily feasible. Since signature verification is the simplest task of all, mobile nodes are able to check a bearer chain on the fly.

The key sizes listed in Table 6.1 are in part very conservative choices for our scenario. While usually RSA keys of at least 1024 bits are recommended [LV01], we can use much shorter keys for our application. It is very unlikely that an attacker would try to break the RSA cryptosystem by factoring on his PC, because the costs (in terms of CPU usage) would prevail over the possible benefit by far. Also users change their key pairs often and the validity period of advertisements is limited, so the damage caused by corrupting a single private key is minimal. Because of these lower security requirements, we consider key lengths between 384 and 768 bits as sufficient for signing the bearer chain entries on a PDA. Even if keys can be broken, a producer does not come to harm since bonus points depend on a concrete purchase.

To sign the advertisement, the producer or vendor must use a key pair with a state-of-the-art bit length (at minimum 1024 bits) for which he has a certificate from an appropriate CA. This is due to the fact that the merchant's signature should provide non-repudiation. Verifying such a signature can be done efficiently on the PDA as is shown in Table 6.1. Doing so, customers can easily learn about the origin of an advertisement and discard unsolicited messages.

6.3.2 Appropriateness of Method

Opportunistic network nodes communicate with nodes in their vicinity without user interaction. Depending on an application's purpose, they reveal personal data of its user that might put user privacy at risk if exploited. For example, adPASS reveals a user's interest in certain product classes. If this information could be uniquely linked to the person, his privacy is lost. In order to make such linkage difficult for an attacker, our privacy preserving strategy features the following characteristics:

- On purpose, the profile is stored solely on a user's personal device. There is no copy present at any server component, as is, for example, the case with the *BlueAware* system by Eagle et al. [Eag05, EP06].

  Thus, the preconditions for an attacker are an active victims node and the attacker being close-by. There is no remote attack possible against our design.

- The opportunistic network *one-hop* communication paradigm allows for a network stack design that omits static network identifiers like fixed MAC or IP addresses. Thus, a node is free to change its network IDs from time to time. This makes linkage from observed profile data to a unique ID harder. Note that even if an attacker would successful link profile data to a unique ID, he still needs to discover the true identity of the user carrying the device.

- The usage of multiple public keys on the application layer as user aliases makes it difficult for an attacker to link different pieces of information that a device gives away to a single user. For example, a node signing a bearer chain (see Section 5.2) could use different keys for different chains. Thus, a malicious Information Producer would not be able to link these chains and construct interest profiles of users.

Please note that these means cannot completely *guarantee* user privacy in opportunistic networks. A hypothetical attacker who is able to mimic several devices, always stays in close proximity to a victim to observe network ID changes, and is equipped with sufficiently powerful hardware to break public keys will be able to break the victim's privacy. Though this kind of attack might be successful, it is very costly compared to the possible benefit. Especially a proximity presence is expensive in terms of human resources and hardware deployment.

6.4 Summary

This chapter presented our results in evaluating the technical feasibility aspects of opportunistic networks with a focus on the proposed data dissemination mechanism and the privacy preserving scheme. For our evaluation, we built two prototypes based on the *iClouds* architecture, namely musicClouds and adPASS. We used the prototypes to conduct real world data dissemination tests and measured runtime behavior of cryptographic functions that are essential for the privacy preserving mechanism. We showed that off-the-shelf hardware is suitable for opportunistic network applications, with battery power being the only limitation.

Chapter 7

Simulation

This chapter presents the second part of our evaluation. We simulated the data dissemination process within an opportunistic network to gain insight into the following questions:

1. How good is the information dissemination coverage, i.e., how many individual information wishes could be fulfilled within one week of simulation?

2. What is the dissemination benefit obtained by deploying Information Sprinklers in the network and, moreover, connecting them?

3. How many hops does it take for an information item to reach a user in different settings?

4. How does the individual user's sharing behavior, i.e., a selfish vs. generous attitude, affect the system's effectiveness?

For all these questions, two different communication ranges were considered: 10 meters to model a device with Bluetooth-like communication capabilities and 100 meters to model a device with WiFi-like communication capabilities, were considered. Our simulator does not take different bandwidths into account.

In addition to the four questions, we compare the effect the different mobility models we used as part of the simulation have on the information dissemination effectiveness. We compared Random Waypoint, Gauss-Markov, and Manhattan Grid mobility models with each other (see Section 7.8).

To answer the questions, a novel *two-step* simulator was developed. Its unique feature is the combination of realistic user mobility data (step one)– we used user traces from the Reality Mining Project [EP05, EP06, Eag05, Mas05] – with commonly used user mobility models (step two). We refer to step one as users' *macro* mobility and to step two as users' *micro* mobility. Our approach remedies the fact that current user mobility models are too simplistic and do not reflect reality well, as argued by Jardosh et al. [JBRAS03] and Bai et al. [BH06]. The next section compares our approach with similar work that also takes user traces into account.

7.1 Related Work

Recently, user traces gained interest among the research community and have been used to study different opportunistic network aspects.

Chaintreau et al. [CHC$^+$06] and Hui et al. [HCS$^+$05] studied the transfer opportunities between mobile devices carried by humans by analyzing several user traces. They found that the distribution of the inter-contact time of a pair of devices, i.e., the time gap between two successive contacts, follows approximately a power law distribution. Phanse and Nykvist [PN06] present a preliminary analysis of 2 user traces with a focus on statistical properties like node degree distribution and topological properties like cluster occurrences.

In [PPC06] an overview of opportunistic message forwarding and routing techniques that take user traces into consideration is given. Opportunistic message forwarding assumes an end-to-end communication need between two or more communication partners but without a direct path between the endpoints. Communicating wirelessly and exploiting node mobility (and sometimes some intermediate infrastructure similar to Information Sprinklers), eventually a delay tolerant duplex link is established.

So far, data dissemination in opportunistic networks has been studied primarily using artificial user mobility models. Becker et al. [BBH02] simulated the performance of epidemic-like diffusion algorithms. For user mobility, they used two artificial models, Random Waypoint and a graph-based mobility model [THB$^+$02]. The authors stress the importance of using realistic mobility models in order to get more realistic simulation results. This is also true for the work of Khelil et al. [KBTR02], who also use Random Waypoint in their simulation. Kathiravelu and Pears [KP06] present a very application specific artificial user mobility model for an airport setting, in order to simulate data distribution among passengers boarding or leaving a plane.

The novelty of our simulation is the combination of user traces with artificial mobility models. Our approach has two advantages. First, it is more realistic than simulations based solely on artificial models, since the traces of real users are involved. Second, our requirements for the user traces are very light. Besides a unique user id, we need to know a distinguishable location ID and a timestamp a user has visited a location. This requirement can be found in very different user traces. For example:

- WiFi MAC addresses seen by WiFi access points.

- Celltower IDs logged on mobile phones.

- Badge IDs logged at doors of a building access (entrance) systems.

- Bluetooth MAC addresses seen by fixed Bluetooth info stations.

- User IDs logged at console login.

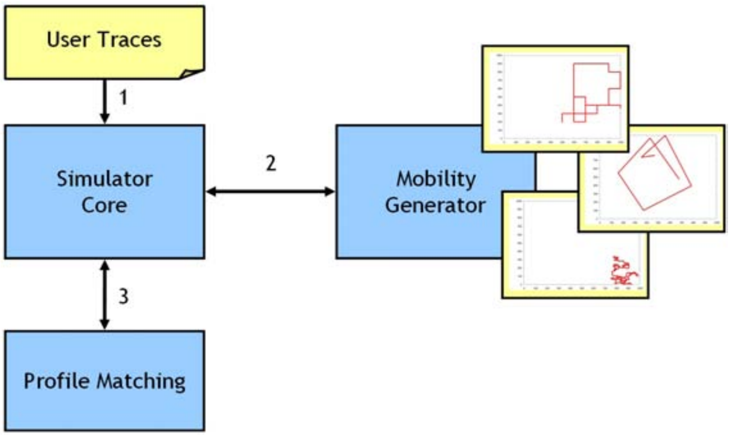


Figure 7.1: Simulator overview

For all these logs, a distinguishable user leaves a mark (of action) at a known location at a certain time. By consolidating these logs, it is easy to query for user pairs that have been seen at the same or a close-by location at a certain time. This is the only information needed for our *micro* mobility step in the simulation. We give details on this fact in the next section.

A number of different traces is publicly available at the CRAWDAD archive at Dartmouth College [Dar05]. As already said, for our simulation we used the Reality Mining Project user traces. We will present our simulation method and the way we include the user traces in the next two sections.

7.2 Overall Simulator Operations

Figure 7.1 shows an simplified vertical architectural view of our simulator. There are three major components (blue boxes): *Simulator Core*, *Mobility Generator*, and *Profile Matching*. The simulator operates as follows:

1. For a given time interval, *Simulator Core* extracts a list of user whereabouts from the Reality Mining data set. Each user whereabout consists of a unique location ID (in our case an ID from a cellular network) and a list of unique user IDs. This information defines a user's mobility in step one, i.e., the *macro* mobility.

2. Each user whereabout is used to generate a logical location (a square with a certain dimension). Users with the same whereabout are assigned to that location and are moved around for a certain time using a certain user mobility model. For the user mobility generation, the BonnMotion [Uni05] mobility generator is used. This step models a user's *micro* mobility.

   Next, the *Simulator Core* gets feedback from the *Mobility Generator* whenever two users come into a preset communication range.

```
+-------+------------------+------------------+-------+----------+
|oid    |starttime         |endtime           |person |celltower |
|       |                  |                  | _oid  | _oid     |
+-------+------------------+------------------+-------+----------+
| 366929|2004-09-20 12:02:15|2004-09-20 12:03:44|     71|        42|
| 732118|2004-09-20 12:03:30|2004-09-20 12:03:51|     58|        41|
|1163074|2004-09-20 12:04:02|2004-09-20 12:04:18|     49|        41|
+-------+------------------+------------------+-------+----------+
```

Figure 7.2: Reality Mining database excerpt. Table `cellspan`

3. *Simulator Core* passes each pair of users that meet each other to the *Profile Matching* component. If, according to the profiles of the users, one user is able to fulfill an information wish of the other user, the corresponding information item is exchanged and a successful information exchange is reported to *Simulator Core*.

At the end of a simulation, several simulation reports are generated by *Simulator Core*. The results are presented and discussed in Section 7.6. At startup, the simulator is parameterized in several ways: Start date (user traces), end date (user traces), increment time interval (user traces), amount of users possessing the information item at the beginning, amount of users with a *free-rider*-behavior, amount of users with a *generous*-behavior (see Section 7.4), used mobility model, use of Information Sprinkler (‘off’,‘on’,‘on and connected’), and communication range between two nodes. The amount of simulated users and the number of locations are determined by the user traces. We get both values from the Reality Mining data set, which we will discuss in the next section.

7.3 Reality Mining Data Set Usage in Simulation

The Reality Mining experiment [Mas05] conducted at Massachusetts Institute of Technology (MIT) Media Lab captured communication, proximity, location, and activity information from 100 subjects (97 useful records) at MIT between 1 January 2004 and 5 May 2005. For this, each subject was given a mobile phone that runs a special software. In general, this software logged communication behavior of the user as well as location information it learned from its surrounding. For our simulation, we used the cellular network tower ID to which a mobile phone was connected at a certain time. Figure 7.2 shows an extract from the Reality Mining data set. Here, on 20 September 2004, within the time interval `[12:00:00 - 12:05:00]`, subject `71` was connected to celltower `42` and subjects `58` and `49` were connected to celltower `41`. From this information we derive that subject `58` and subject `49` were co-located at that time interval. As said in the introduction of this chapter, this information is used for the *micro* mobility step in the simulation process. Considering subjects `58` and `49`, they are put on a virtual square (size 1000 × 1000 meters) and moved around for 5 minutes (the time interval span)

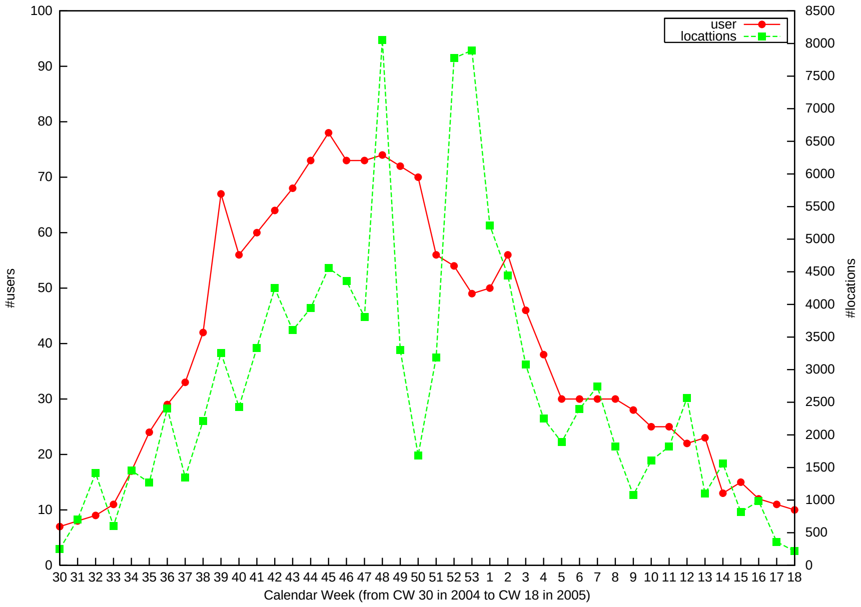


Figure 7.3: User and location logs (Reality Mining data set)

according to a given mobility model. This allows the simulator to detect the distance between subject 58 and 49. If they come close enough to each other to be in a given communication range, i.e., 10 meters for the Bluetooth setting and 100 meters for the WiFi setting, the Simulator executes the profile matching algorithm between them. On a match, information between subject 58 and 49 is exchanged.

The Reality Mining experiment was run for over 16 months and the density of logged user traces varies a lot between individual weeks. Very few traces were logged before calendar week 30 in 2004. Starting from calendar week 30, both the number of logged user IDs and the number of logged celltower IDs increase. Figure 7.3 depicts both curves. For our simulation, three different weeks were chosen. Calender week 45 with the highest amount of active users (78), calender week 48 with a fairly high number of users (74) and a maximum number of logged celltower IDs (8056), and calender week 50, again with a fairly high amount of active users (70) and a very low number of logged celltower IDs (1685). Calender week 48, lasting from 22 November 2004 to 29 November 2004, recorded a very large number of different celltower IDs, since Thanksgiving (25 November 2004) fell in that week. We assume the subjects left MIT to visit family and friends and thus more different celltower IDs where logged during that week. Most likely the same holds for calender week 52 and 53, the Christmas holidays. Table 7.1 summarize our calendar week selection.

	Week 45	Week 48	Week 50
#active users	78	74	70
#locations	4557	8056	1685

Table 7.1: Chosen calendar weeks in 2004

7.4 User Behavior

Within the simulation, we distinguish between two types of users. One user-type, labeled *free-rider*, acts purely selfish and does not share any information he collects. The other user-type, labeled *generous*, shares any information he collects with others. For the simulation results presented in Section 7.6, all users act *generous*. Later, we vary the *generous/free-rider* ratio to see how individual user behavior affects the overall dissemination effectiveness. Results are presented in Section 7.7.

7.5 Mobility Models Used

The simulation uses three different mobility models for the users' *micro* mobility. For the data dissemination process (results are shown in Figures 7.4, 7.5, 7.6, 7.7, 7.8, and 7.9), the Random Waypoint [CBD02] mobility model was employed. Random Waypoint is simple and often used [BMJ+98, CG98, GLAS99, JLH+99]. Within this model, a mobile node begins by staying at one location for a certain period of time, the so-called *pause time*. Once the pause time expires, the mobile node chooses a random destination in the simulation area and a speed. This speed is uniformly distributed between $[minspeed, maxspeed]$. The mobile node moves towards the newly chosen destination at the chosen speed. Upon arrival, the mobile node pauses for a new randomly chosen pause time before starting the process again.

The Manhattan Grid [Eur98] mobility model and the Gauss-Markov [CBD02] mobility model were used to review the data dissemination results, since the *micro* mobility model affects the chance of users coming into communication range and thus the chance to exchange information at all.

In the Manhattan Grid model, mobile nodes move only on predefined horizontally and vertically arranged paths. This model mimics a typical street network in an urban area. A mobile node starts at a randomly selected position on a path, chooses its speed between $[minspeed, maxspeed]$ and direction and moves along a path. Periodically the chosen speed is adjusted. In addition, a node may pause for a certain time or turn its direction at a crossing.

The Gauss-Markov Model eliminates sudden stops and sharp turns encountered in the Random Waypoint Mobility Model by allowing past velocities and directions to influence future velocities and directions. For this, a mobile node is assigned a certain speed and direction. Periodically new values for speed and direction are chosen from a normal distribution with a mean of the respective old value. Speed values are constrained to a certain interval. If a newly chosen speed value is outside this interval, it is reset to the closed value inside the interval.

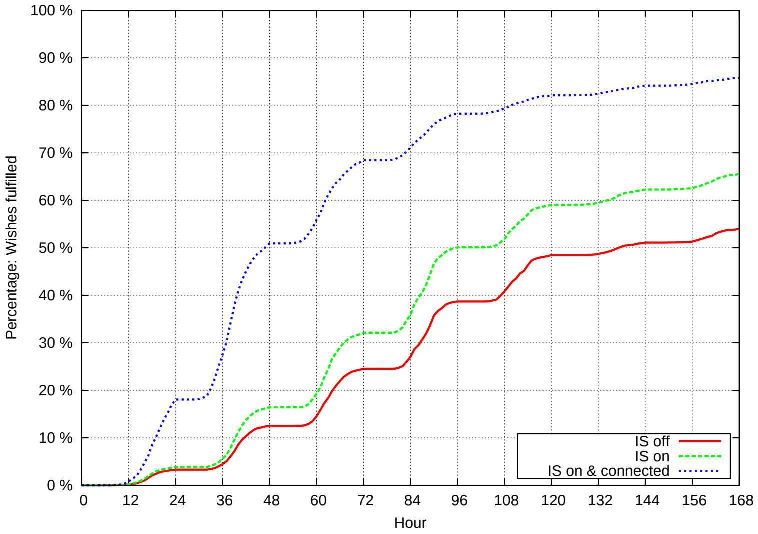
The concrete settings we used for the different mobility models during simulation are given in Section 7.8.

7.6 Simulation Results: Dissemination Effectiveness

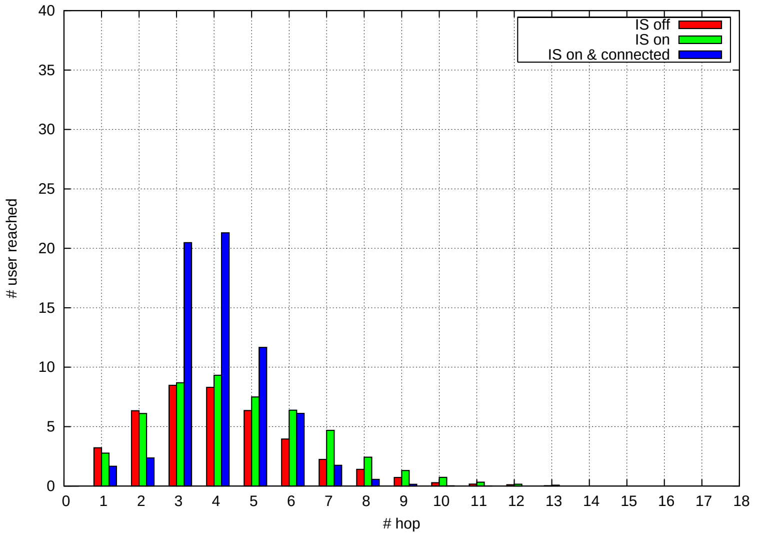
This section presents several simulation results for calender week 45, 48, and 50 considering 10 meters and 100 meters as a user's device communication range. We refer to a 10 meter communication range as *Bluetooth* scenario and to a 100 meter communication range as a *WiFi* scenario. For each week, the simulation was run 100 times (the figures display the averaged values) and with three different setups. In setup one, there are no Information Sprinklers to help with *time shifted* data dissemination (see Section 3.4.2). In setup two there are Information Sprinklers in place, and finally in setup tree, all Information Sprinklers are connected by a backbone network (see page 36). In this last case, as soon as an information item is passed from a user to an Information Sprinkler, this information item is available at all other Information Sprinklers at all other locations. For all runs, the user behavior was set up as follows. One user, chosen randomly, owns an information at startup. All other user are interested in the information and all users acted generously, i.e., they always pass the information on to others.

Figure 7.4 shows the simulation results for calender week 45, Bluetooth scenario. Figure 7.4(a) shows the amount of overall fulfilled wishes observed at simulation time (broken down to hours). Without any Information Sprinklers in place, at the end of a simulated week, on average 53.97% of wishes are fulfilled. Deploying an Information Sprinklers at each location, this value increases to 65.49%. Finally, connecting all deployed Information Sprinklers to a backbone network increases the amount of overall fulfilled wishes to 85.77%.

Looking at the number of hops, i.e., opportunistic network nodes, the information travels within one week of simulation (depict in Figure 7.4(b)), most users are reached with either 3 or 4 hops. The maximum hop count observed is 12. Deploying Information Sprinklers most users are reached with 4 hops. Starting from 3 hops, the number of hops increases slightly, extending the maximum hop count observed to 13. Different from this, connecting the Information Sprinklers to a backbone network significantly increases the number of users reached with 3,4, and 5 hops and reduces the maximal observed hop count to 9. The reason for a jump in reached users from 2 hops to 3 hops is as follows: User A_i passes the information at a location loc_l to an Information Sprinkler IS_r (counts as 1 hop), next the information is synced to all other Information Sprinklers in the network (counts as 1 hop). Finally, at any other location loc_m the information is passed from an Information Sprinkler IS_s to another user A_j (counts as 1 hop). Thus, Information Sprinklers connected to a backbone have two effects. Many information wishes are fulfilled by information

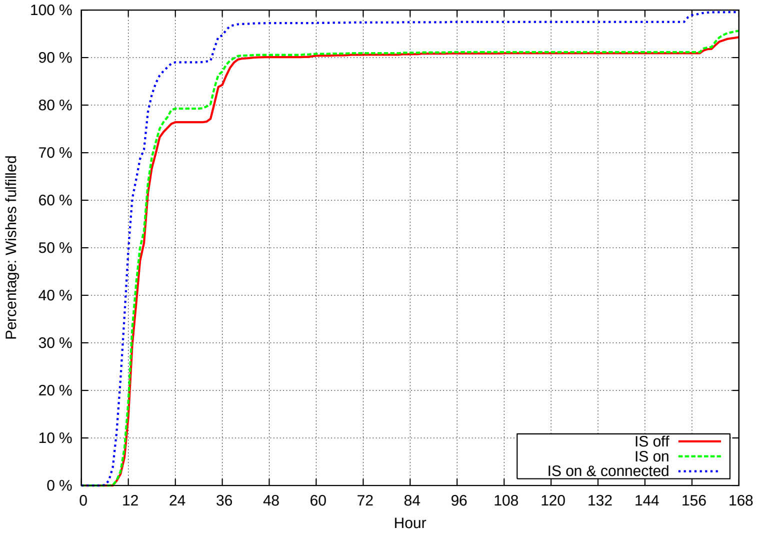


(a) Hour/fulfilled wishes ratio

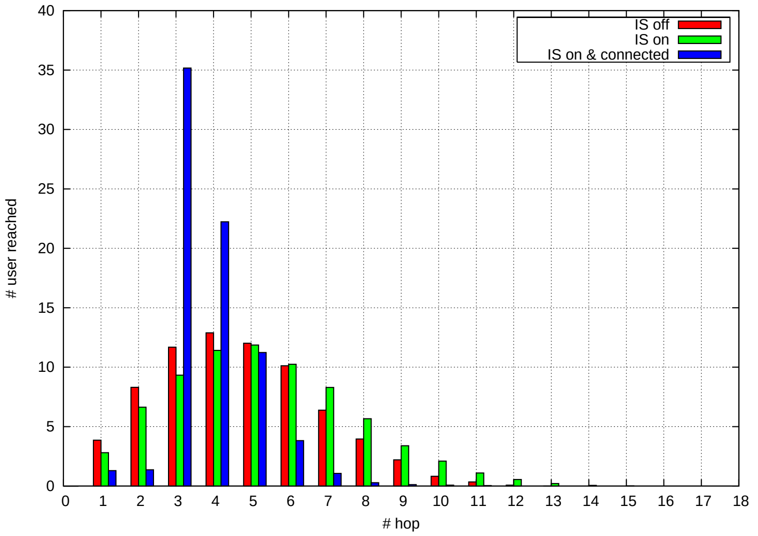


(b) Hops/reached users ratio

Figure 7.4: Calendar week 45 - Bluetooth scenario



(a) Hour/fulfilled wishes ratio



(b) Hops/reached users ratio

Figure 7.5: Calendar week 45 - WiFi scenario

that is provided by a deployed Information Sprinkler and second, the maximum number of hops observed is reduced.

Considering a device communication range of 100 meters, i.e., a WiFi scenario, the amount of fulfilled wishes observed is quite different. Figure 7.5(a) depicts that already without any Information Sprinkler in place, 94.30% of the users' information wishes are fulfilled after one week. Next, there is not much effect from deploying Information Sprinklers at each location. At the end of the week, 95.62% of wishes are fulfilled. Connecting all Information Sprinklers increases the amount of fulfilled wishes to 99.58%. Looking at Figure 7.5(b), which depicts the number of hops it take for an information to reach users, the behavior resembles the observed results in the Bluetooth scenario, although the absolute numbers are higher.

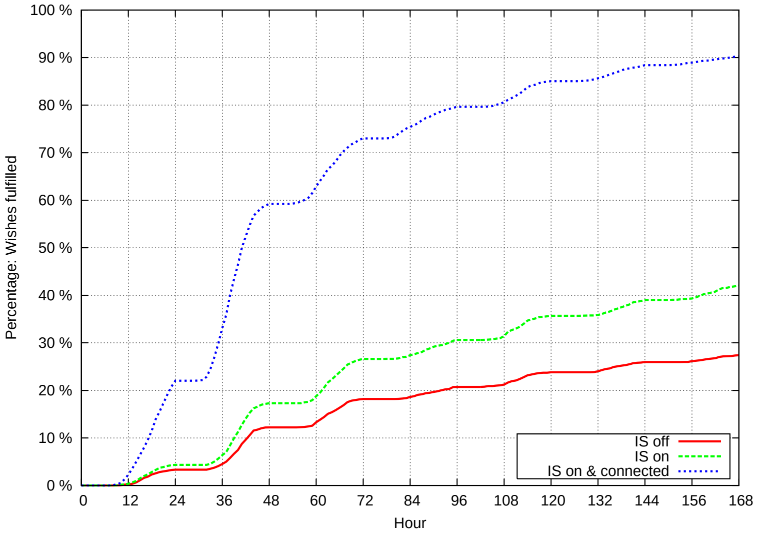
Figures 7.7 and 7.9 present the simulation results for calender weeks 48 and 50, respectively. Although the number of logged locations is close to twice as much in calender week 48 (8056) and less than half as much in calender week 50 (1685), the results are comparable to calender week 45. For the Bluetooth scenario, both weeks see a significant increase in the overall number of fulfilled wishes when deploying and connecting Information Sprinklers and an about 10% increase by just putting Information Sprinklers in place. For the WiFi scenario, deploying Information Sprinklers (connected or not) does not have much effect on the overall information dissemination performance.

7.7 Simulation Results: Different User Behavior

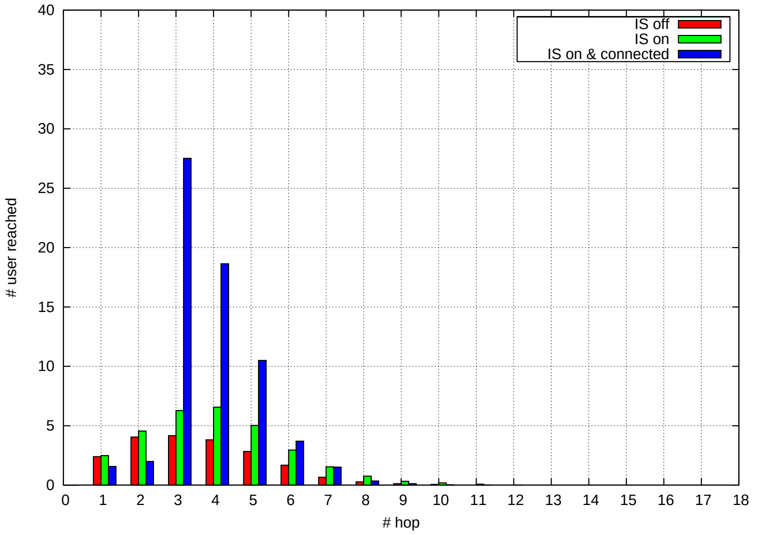
In order to gain insight into how the users' behaviors affect the information dissemination process, we used the calendar week 45 user traces with no Information Sprinklers in place, the Random Waypoint mobility model for the users' *micro* mobility and considered 10 meters and 100 meters as communication ranges. Same as before, one information item is randomly assigned to one user at startup. All other users are interested in the information. But this time, only a fraction of the users act generously. Other users act selfishly, i.e., as free-riders. They take the information from another user but do not distribute the information further.

Figure 7.10(a) shows the results in a Bluetooth scenario and Figure 7.10(b) in a WiFi scenario. We simulated three different free-rider/generous ratios: 20% free-riders / 80% generous, 50% free-riders / 50% generous, and 80% free-riders / 20% generous. For reference, the figures also show the case with all users acting generous. With an 20% free-riders / 80% generous ratio, the amount of fulfilled wishes drops from 53.97 % to 46.06 %. In the 50% free-riders / 50% generous it drops further to 23.38 % and falls to 9.80 % in the 80% free-riders / 20% generous ratio case. Thus, the amount of free-riders harms the dissemination process a lot if we consider a 10 meter communication range.

In the WiFi scenario, we obtain different results. Here, the 80% free-riders / 20% generous and 50% free-riders / 50% generous ratio distribution in user behavior do not vary significantly from the 100% generous users case (see Figure 7.10(b)).

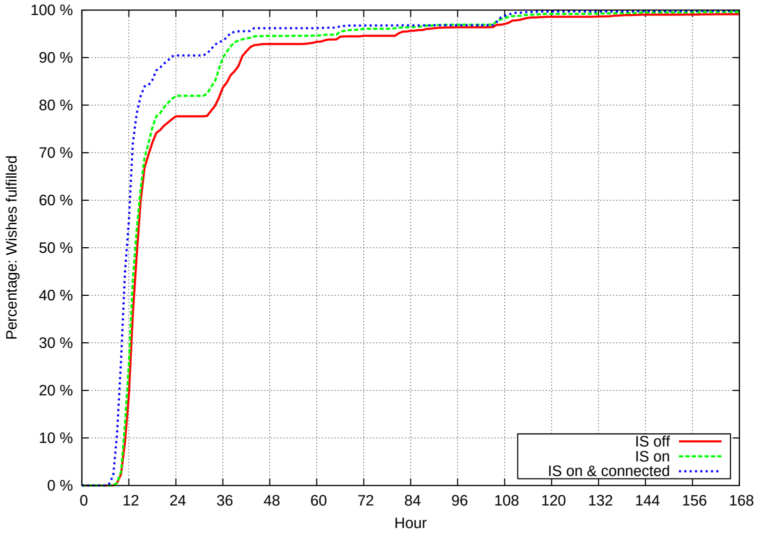


(a) Hour/fulfilled wishes ratio

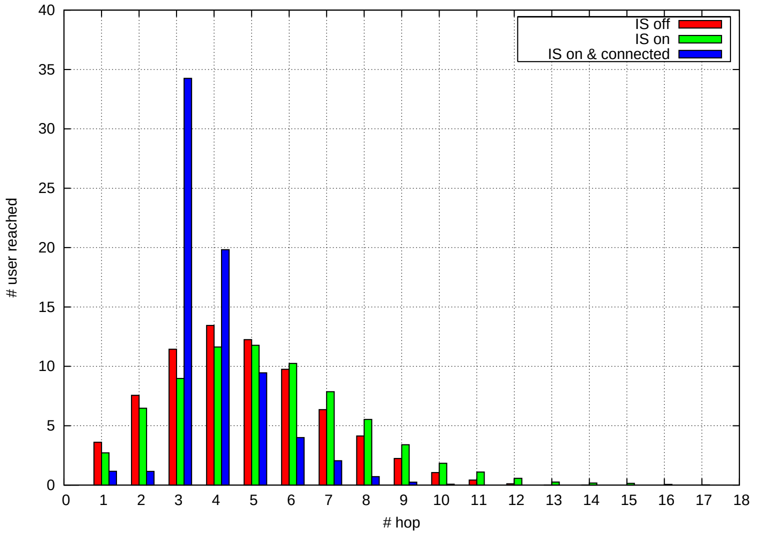


(b) Hops/reached users ratio

Figure 7.6: Calendar week 48 - Bluetooth scenario

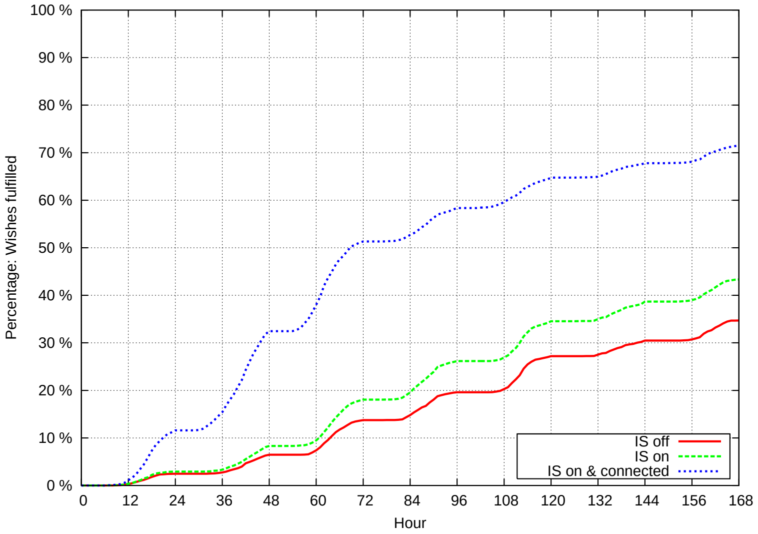


(a) Hour/fulfilled wishes ratio

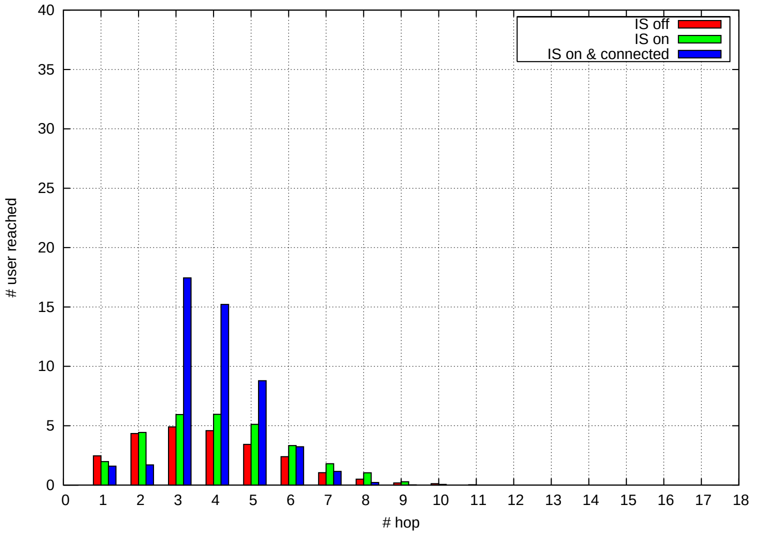


(b) Hops/reached users ratio

Figure 7.7: Calendar week 48 - WiFi scenario

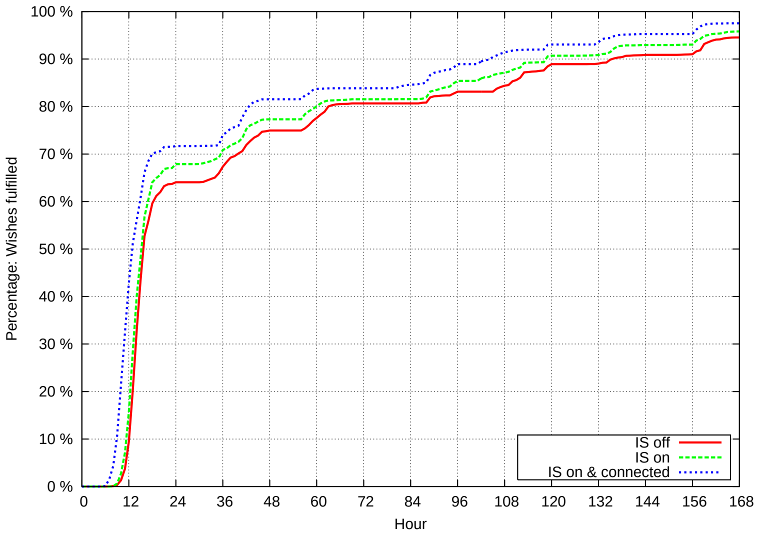


(a) Hour/fulfilled wishes ratio

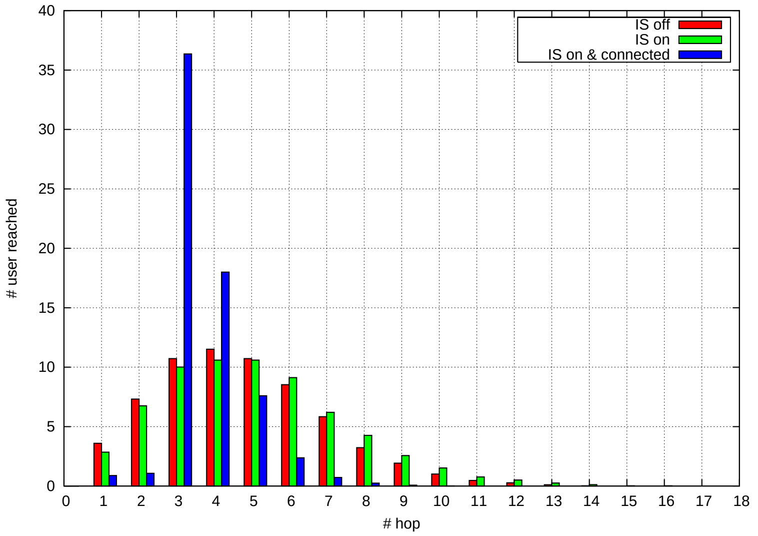


(b) Hops/reached users ratio

Figure 7.8: Calendar week 50 - Bluetooth scenario



(a) Hour/fulfilled wishes ratio



(b) Hops/reached users ratio

Figure 7.9: Calendar week 50 - WiFi scenario

	Random Waypoint	Gauss-Markov	Manhattan Grid
Dimension	1000 × 1000 m	1000 × 1000 m	1000 × 1000 m
#blocks (x/y)-axis	n/a	n/a	10/10
Update distance	n/a	n/a	5 m
Turn probability	n/a	n/a	0.5
Update frequency	n/a	2.5 s	n/a
Mean speed	n/a	n/a	1.0 m/s
Min. speed	0 m/s	0.25 m/s	0.5 m/s
Max. speed	0.5 m/s	2.0 m/s	n/a
Max. pause	180 s	n/a	120 s

n/a = not applicable

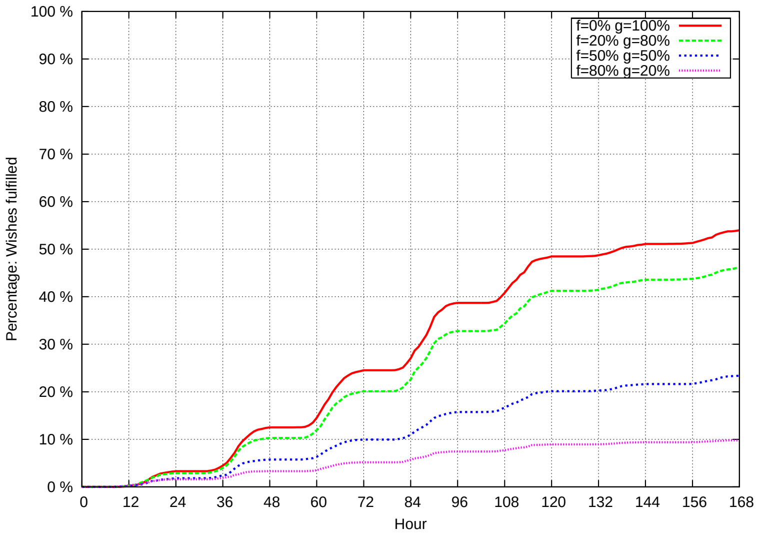
Table 7.2: Used mobility models and settings

Only in the 80% free-riders / 20% generous ratio case does the overall amount of fulfilled wishes after one week of simulation drop to 84.08%. Thus, a 100 meter communication range helps to alleviate the effect introduced by the selfish behavior of free-riders.

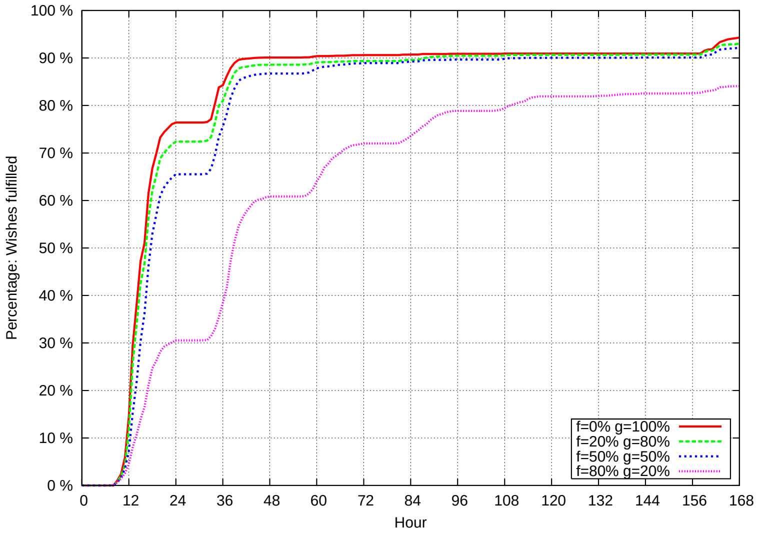
7.8 Simulation Results: Different Mobility Models

There are two premises for users to encounter each other. First, the simulator determines co-located users by making use of the Reality Mining data set. Second, the *micro* mobility of co-located users is simulated using a synthetic model. Only users that come into a given communication range to each other are able to exchange information. Until now, for all obtained simulation results the Random Waypoint mobility model was used in simulation step two. Obviously, the chosen mobility model affects the likelihood of users to come into communication range and thus affects a mandatory pre-condition for the data dissemination process. We chose two other mobility models, namely Gauss-Markov and Manhattan Grid, to figure out the *micro* mobility model influence on the data dissemination process. Table 7.2 summarizes the parameter settings for the applied mobility models. Again, we choose calendar week 45 for step one in the simulation without Information Sprinkler deployment. The averaged results from 100 simulation runs are shown in Figure 7.11.

Figure 7.11(a) shows that Gauss-Markov and Manhattan Grid yield a better data dissemination process than Random Waypoint in a Bluetooth scenario. With Gauss-Markov, 92.59% of the wishes are fulfilled after one week of simulation, Manhattan Grid results in 96.58% fulfilled wishes. These are significantly better results than with the Random Waypoint Model (53.97% fulfilled wishes).

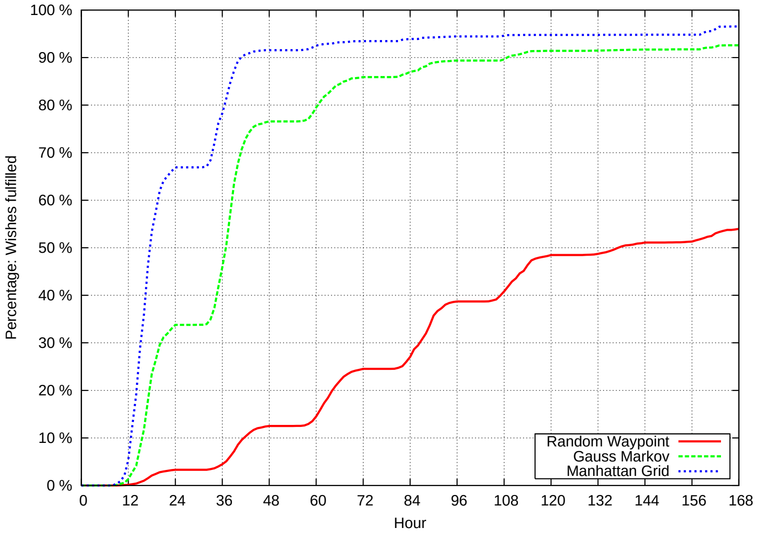


(a) Bluetooth scenario

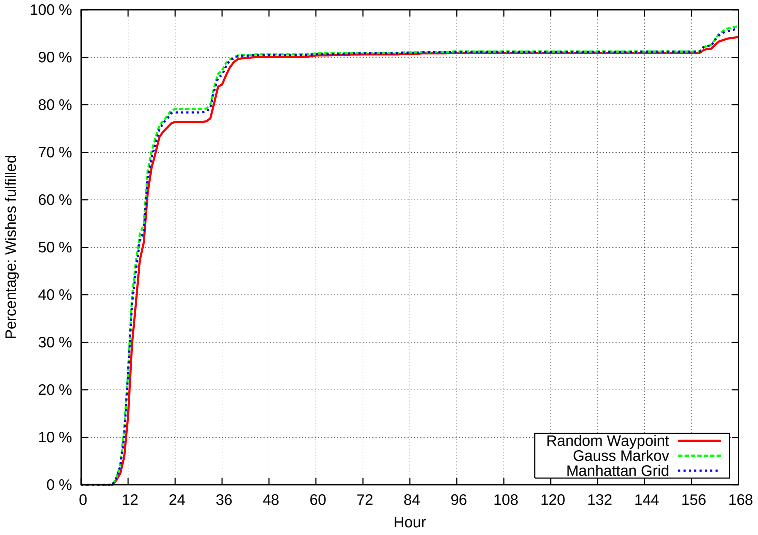


(b) WiFi scenario

Figure 7.10: Comparing various free-rider/generous ratios (Calendar week 45, Information Sprinklers off)



(a) Bluetooth scenario



(b) WiFi scenario

Figure 7.11: Comparing mobility models (Calendar week 45, Information Sprinklers off)

The picture is different in the WiFi Scenario, see Figure 7.11(b). Here, there is no notable difference in the data dissemination process efficiency. Gaus-Markov yields 96.51% fulfilled wishes, Manhattan Grid yields 96.13% fulfilled wishes and Random Waypoint with 94.30% fulfilled wishes performed slightly worse.

7.9 Discussion

This section discusses the simulation results and their meanings based on our underlying model and used user traces from the Reality Mining data set. As in every model, our model simplifies the real world. The total number of users we consider is fairly low, i.e., 97 recorded users from the Reality Mining data set. Next, besides the celltower ID hint, we do not know anything about their whereabouts and consequently put them on a 1000×1000 square to move them around by a synthetic mobility model. Both settings, the 1000×1000 square and the synthetic mobility model, are probably inaccurate in capturing the real world. Nonetheless, this approach is better than using synthetic mobility models alone as commonly practiced.

Also, we assume that every user is interested in the information. In a real world, different users will have different interests. But even then, it is possible to form a set of all users interested in one particular information item. Thus, we are looking at the dissemination process of this one information item. Finally, the users are not randomly spread over a country but have a connection to MIT (students, faculty members, etc.). Thus, the user set is not totally unrelated and might not even be totally anonymous to each other. On the other hand, one goal of opportunistic network applications is to work specifically in those settings.

Bearing the limitations of our model in mind, we discuss the results by looking at each question raised in the introduction of this chapter in turn.

1. How good is the information dissemination coverage, i.e., how many individual information wishes could be fulfilled within one week of simulation?

   Bluetooth scenario: Assuming a 10 meter communication range, the coverage varies a lot. It ranges from ≈27% to ≈91%, with a significant improvement if Information Sprinklers are deployed and connected.

   WiFi Scenario: Assuming a 100 meter communication range of an opportunistic network device, the coverage is between ≈94% and ≈99%.

2. What is the dissemination benefit obtained by deploying Information Sprinklers in the network and moreover connecting them?

   Bluetooth scenario: The dissemination efficiency benefits from deploying Information Sprinklers by ≈10% of more fulfilled wishes. The benefit is even higher if the Information Sprinklers are connected by a backbone network. Here the percentage of fulfilled wishes ranges from ≈72% to ≈91%.

WiFi scenario: The benefit of deploying Information Sprinklers (connected or not) does not improve the efficiency much. But, with their help, 100% of the wishes in calendar week 45 and calendar week 48 could be fulfilled.

3. How many hops does it take for an information item to reach a user in different settings?

   The maximum number of hops it took to reach a user was 17. The majority of users were reached within 2 – 7 hops in both scenarios (Bluetooth and WiFi). This information might be considered for an Information Producer in the adPASS incentive scheme (see Section 5.2.1 in Chapter 5).

4. How does an individual user's sharing behavior, i.e., a selfish vs. generous attitude, affect the system effectiveness?

   Bluetooth scenario: Here, the individual user's behavior is critical for the dissemination effectiveness. The number of fulfilled wishes drop to ≈10% if a 80% free-rider, 20%generous population is assumed, with is typical for Internet based Peer-to-Peer networks and might be true for opportunistic networks as well.

   WiFi scenario: Due to the better communication range, an increasingly amount of free-riders has less effect on the dissemination effectiveness. Looking at the 80% free-rider / 20%generous population, the overall percentage of fulfilled wishes drops by only ≈10%.

Altogether, looking at our simulation results, a communication range of 100 meters remedies the otherwise negative impact on the dissemination effectiveness of free-riders in the network. This makes incentive schemes such as proposed in adPASS less critical for the application acceptance. The deployment of an Information Sprinkler backbone, which might be cost-intensive, is not necessary. However, a wider communication range makes opportunistic networks applications in the *active collaboration* domain less useful since it will be harder for users to recognize each other after a successful profile match. Also, proximity based services lose their location accuracy with a wider communication range. For both, a better accuracy and a good dissemination efficiency, a 10 meter communication range combined with an incentive scheme combined with an Information Sprinkler backbone network is the better choice.

To summarize, when choosing the communication rage (by selecting a target device) for an opportunistic network application, a designer or developer has to keep the pros and cons of a wider communication range in mind and select the most suitable approach for his application.

7.10 Summary

This chapter presented our opportunistic network simulator and various results evaluating the effectiveness of our proposed data dissemination mechanism. The simulator novelty is the combination of real world user traces with artificial mobility models. This two step approach is suitable for a number of different collected user traces due to the few requirements the user traces need to fulfill.

Within the scope of our simulation model and assumptions, the results reveal promising insight into the effectiveness of the data dissemination mechanism under various settings and help application developers to understand the impact of communication range, user behavior and preset infrastructure deployments on the data dissemination effectiveness.

Chapter 8

Summary and Outlook

This chapter concludes this thesis by summarizing the issues addressed and major findings. Then we provide an outlook on open questions in this field and give hints for future research directions.

8.1 Summary

The integration of wireless, short-range communication capabilities in personal mobile devices paves the way for opportunistic networks and their applications. Opportunistic network applications expose several characteristics and ideas like the exploitation of user's vicinity, user profile-based interest expression, autonomous dissemination of information, an unpredictable communication pattern, and an open and unrelated user group. Current research addresses these ideas or a subset of these ideas heterogeneously. Most work overlooks the human aspects of opportunistic networks and its applications. The thesis addresses opportunistic networks in its entirety. We identified privacy issues and incentives as two crucial human aspects for users' acceptance of opportunistic networks. For both issues, solutions have been given in this work.

This thesis shows that human aspects have an impact on the technical tasks in opportunistic networks. Our one-hop communication paradigm between directly connected wireless nodes makes the notion of dynamically self generated user identifiers feasible in the first place. The combination preserves user privacy in a simple and elegant way.

Furthermore, our opportunistic network system model includes support for proximity based services, that are easily and deployable in a decentral manner and do not need any pre-installed infrastructure. This is possible by putting up Information Sprinklers at dedicated locations.

Next, a formal and thus technology independent model for user interest expression via profiles and the task of matching user profiles against each other as a basis for our data dissemination process is provided as part of this thesis. This model and its provided algorithms in pseudo-code will help future work to clarify and sharpen further the opportunistic networks domain.

Incentives, as the second important human aspect of opportunistic networks, are addressed within this work. Our incentive scheme resides on top of our core opportunistic network concepts that aims to stimulate users to participate in an opportunistic network application without harming a user's privacy. The adPASS prototype, developed as part of this thesis, fully implements the incentive scheme and opportunistic network concepts.

We addressed opportunistic network application technical feasibility by implementing two prototypes and running several real world tests. These tests measured data throughput on real devices. In addition, runtime behavior of cryptographic functions that are essential to preserve a user's privacy and the incentive scheme in general were measured. We showed that off-the-shelf hardware is already powerful enough for opportunistic network applications.

The effectiveness of the opportunistic network data dissemination process was shown by a novel two-step simulation approach, which combines real world user traces with synthetic mobility models. This combination yields a solid notion of user whereabouts, which is a mandatory information to simulate the data dissemination process. Within this work, we tested various settings and – within our model and assumptions – successfully demonstrated the effectiveness of the data dissemination.

In summary, this thesis

1. identified a number of common criteria that are appropriate for addressing opportunistic networks in their entirety (Chapter 2).

2. defined a model for opportunistic networks, suitable components and communication pattern to serve our privacy preserving mechanisms and incentive scheme. (Chapter 3).

3. defined a general model for data dissemination in opportunistic networks (Chapter 4).

4. provided a solution for user privacy preservation and an incentive scheme (Chapter 5).

5. showed that off-the-shelf hardware is powerful enough to build full-featured opportunistic network prototypes (Chapter 6).

6. showed that opportunistic network data dissemination is efficient in various settings using a two-step simulation model (Chapter 7).

Nonetheless, this thesis is not an exhaustive investigation of opportunistic networks. There are many interesting research questions left for further research. We will outline some in the next section.

8.2 Outlook

Opportunistic networks are formed by users that are a priori unrelated and anonymous to each other. This leads to the interesting question of user reputation and trust. For example, within opportunistic network applications like musicClouds, there are no hints if an offered music file is accurate, fulfills the given specification (size, codec, sample rate, etc.), is not faked or damaged, and is virus free. This uncertainty could be solved by introducing user reputation and trust concepts. Here, a user's former action within the network is rated by others and a user is able to build up a certain reputation. This reputation value may be considered by other users before getting involved with that user. Work in this direction has already begun. Voss et al. describes a privacy preserving reputation system for opportunistic networks. The authors take the *iClouds* architecture as a basis for their system. See [VHM05] for details.

Further, reputation may serve as a basis for trust. As stated by Sabater and Sierra [SS05], "[o]ur perspective is that reputation is one of the elements that helps to build trust on others". Thus, user reputation and trust could increase the usefulness of opportunistic network applications even further. A simple approach would be to share information only with trusted nodes or with nodes with a good reputation.

Two issues not covered within this thesis are power and memory management. Although current off-the-shelf mobile devices that are suitable for opportunistic network applications, for example PDAs or mobile phones, become more powerful and are equipped with more memory with every new generation, power and memory consumption may remain an issue to solve depending on the application. An opportunistic network application that shares a movie collection similar to musicClouds is currently not practical. And since the opportunistic network node are primarily personal devices, sufficient power and memory should always be left for personal use like making a phone call. Consequently, opportunistic network applications need mechanisms for efficient power and memory management.

In order to better estimate the data dissemination process, our simulation and model could be improved in several ways. First, more user traces from other cellular networks could be used as input data. Second, the *micro* mobility model could make more realistic assumption, for example, radio inferences introduced by obstacles could be included. Third, a survey of user behavior in Peer-to-Peer systems could be used to estimate the free-rider/generous ratio with higher accuracy.

The developed simulator might be generalized and could then be useful for related tasks. We just need to detach the *two users meet* event from the *start sharing information* action. This would allow us to exchange the kind of action. An example would be to share trust values as proposed by Voss et al. [VHM05].

As stated in Section 5.2.13 of Chapter 5, the adPASS incentive scheme is flexible enough to be applied to other applications that follow a kind of *store* and *forward* mechanism. It would be interesting to see if there are more application domains where an incentive scheme use makes sense. For example, an e-mail based recommendation system for advertisements found on the web is thinkable and, leaving aside privacy issues, is sufficiently similar for implementation.

Finally, an interesting question is how to integrate opportunistic network applications with users' desktop PC usage. Currently, a user has to enter his information interests by hand on the device and actively put the information he wants to share with others on the device. By monitoring a user's Internet usage, i.e., what kind of queries he sends to Google or what kind of web sites he visits on a regular basis, it might be possible to derive his interests and automatically put entries on the iWish-list. Similar, PC to handheld device synchronization mechanisms could be applied to automatically fill a user's iHave-list.

Bibliography

[ABCG03] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. IEEE 802.11 Ad Hoc Networks: Performance Measurements. In *23rd International Conference on Distributed Computing Systems Workshops (ICDCS 2003 Workshops)*, pages 7580–7586. IEEE Computer Society, 2003.

[ABCG04] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Wi-Fi in Ad Hoc Mode: A Measurement Study. In *Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 145–155. IEEE Computer Society, 2004.

[AG04] Kevin C. Almeroth and Anargyros Garyfalos. Coupons: Wide Scale Information Distribution for Wireless Ad Hoc Networks. In *IEEE Global Telecommunications Conference (Globecom) Global Internet and Next Generation Networks Symposium*, pages 1655–1659. IEEE Computer Society, 2004.

[AGKO04] Lauri Aalto, Nicklas Göthlin, Jani Korhonen, and Timo Ojala. Bluetooth and Wap Push Based Location-Aware Mobile Advertising System. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 49–58. ACM Press, 2004.

[AHrg] Eytan Adar and Bernardo A. Huberman. Free Riding on Gnutella. *First Monday*, 5(10), 2000. `http://www.firstmonday.org/`.

[Ama95] Amazon.com Inc. amazon.de [online]. 1995. Available from: `http://www.amazon.de` [seen 2007].

[ANG02] Roberto Aldunate, Miguel Nussbaum, and Roberto Gonzalez. An Agent-Based Middleware for Supporting Spontaneous Collaboration among Co-Located, Mobile, and not necessarily Known People. In *Workshop on 'Ad hoc Communications and Collaboration in Ubiquitous Computing Environments',Conference on Computer Supported Cooperative Work (CSCW)*. ACM Press, 2002.

[App03] Apple, Inc. iTunes [online]. 2003. Available from: http://www.apple.com/itunes/ [seen 2007].

[BBH02] Christian Becker, Martin Bauer, and Jörg Hähner. Usenet-on-the-fly – Supporting Locality of Information in Spontaneous Networking Environments. In Ramiro Liscano and Gerd Kortuem, editors, *Workshop on Ad Hoc Communications and Collaboration in Ubiquitous Computing Environments*, New Orleans, USA, 2002. ACM Press.

[Bea05] Russell Beale. Supporting Social Interaction with Smart Phones. *IEEE Pervasive Computing*, 4(2):35–41, 2005.

[BH06] Fan Bai and Ahmed Helmy. *Wireless Ad-Hoc and Sensor Networks*, chapter A SURVEY OF MOBILITY MODELS. Springer, 2006.

[BLMR98] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *ACM SIGCOMM*, pages 56–67. ACM Press, 1998.

[Blu03] Bluetooth SIG Inc. The Official Bluetooth Membership Site [online]. 2003. Available from: http://www.bluetooth.org [seen 2007].

[Blu05] Bluetooth SIG Inc. Bluetooth Shipments Climb to Five Million Per Week [online]. 24.05.2005. Available from: http://bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Shipments_Climb_to_Five_Million_Per_Week.htm [seen 2007].

[BMJ+98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, pages 85–97. ACM Press, 1998.

[BS03] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Proceedings*, pages 319–327, 1988.

[CG98] Ching-Chuan Chiang and Mario Gerla. On-Demand Multicast in Mobile Wireless Networks. In *1998 International Conference on Network Protocols (ICNP '98)*, pages 262–270. IEEE Computer Society, 1998.

[CGKÖ04] Jon Crowcroft, Richard J. Gibbens, Frank P. Kelly, and Sven Östring. Modeling incentives for collaboration in mobile ad hoc networks. *Performance Evaluation - Selected Papers from the First Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'2003)*, 57(4):427–439, 2004.

[Cha81] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha83] David Chaum. Blind signatures for untraceable payments. In Alan T. Sherman David Chaum, Ronald L. Rivest, editor, *Advances in Cryptology: Proceedings of CRYPTO '82. Plemum*, pages 199–203. Springer, 1983.

[Cha85] David Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[CHC+06] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. In *IEEE INFOCOM 2006*. IEEE Computer Society, 2006.

[Cli01] Clip2. The Gnutella Protocol Specification v0.4 [online]. 2001. Available from: `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf` [seen 2007].

[CM99] Scott Corson and Joseph Macker. Rfc 2501 mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations [online]. 1999. Available from: `http://www.ietf.org/rfc/rfc2501.txt` [seen 2007].

[Com07] Internet Community. DMOZ – Open Directory Project [online]. 1998-2007. Available from: `http://dmoz.org/` [seen 2007].

[Dar05] Dartmouth College. CRAWDAD Project: A Community Resource for Archiving Wireless Data At Dartmouth. `http://crawdad.cs.dartmouth.edu/` (seen 02/2007), 2005.

[Dat03] Anwitaman Datta. MobiGrid: Peer-to-Peer Overlay and Mobile Ad-Hoc Network Rendezvous – a Data Management Perspective. In *CAiSE 2003 Doctoral Symposium, in conjunction with the 15th Conference On Advanced Information Systems Engineering*. Springer, 2003.

[DB03] Thomas D'Roza and George Bilchev. An overview of location-based services. *BT Technology Journal*, 21(1):20–27, 2003.

[DB04] Gang Ding and Bharat Bhargava. Peer-to-peer File-sharing over Mobile Ad hoc Networks. In *2nd IEEE International Conference on Pervasive Computing and Communications – Workshop on Mobile Peer-to-Peer Computing*. IEEE Computer Society, 2004.

[DGH+87] Alan J. Demers, Daniel H. Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard E. Sturgis, Daniel C. Swinehart, and Douglas B. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, pages 1–12. ACM Press, 1987.

[DM04] Roger Dingledine and Nick Mathewson. Tor: An anonymous Internet communication system [online]. 2004. Available from: `http://tor.eff.org` [seen 2007].

[Dou02] John R. Douceur. The Sybil Attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems: First International Workshop*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

[DQA04] Anwitaman Datta, Silvia Quarteroni, and Karl Aberer. Autonomous Gossiping: A Self-Organizing Epidemic Algorithm for Selective Information Dissemination in Wireless Mobile Ad-Hoc Networks. *Lecture Notes in Computer Science*, 3226:126–143, 2004.

[Eag05] Nathan Norfleet Eagle. *Machine Perception and Learning of Complex Social Systems*. PhD thesis, Massachusetts Institute of Technology, 2005.

[eBa95] eBay Inc. ebay [online]. 1995. Available from: `http://www.ebay.com` [seen 2007].

[EP05] Nathan Eagle and Alex Pentland. Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, 4(2):28–34, 2005.

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

[ESN06] Sebastián Echeverría, Raúl Santelices, and Miguel Nussbaum. Comparative Analysis of Ad-Hoc Networks Oriented to Collaborative Activities. In *Architecture of Computing Systems - ARCS 2006: 19th International Conference, Frankfurt/Main, Germany, March 13-16, 2006. Proceedings*, volume 3894 of *Lecture Notes in Computer Science*, pages 465–479. Springer, 2006.

[Eur98] European Telecommunications Standards Institute. UMTS: Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS. Technical Report TR 101 112 V3.2.0, European Telecommunications Standards Institute, 1998.

[FJP96] Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann. MIXes in Mobile Communication Systems: Location Management with Privacy. In *Information Hiding*, pages 121–135, 1996.

[FP00] Justin Frankel and Tom Pepper. Gnutella Protocol Specs Version 0.4 [online]. 2000. Available from: `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf` [seen 2007].

[Fre04] Hannes Frey. Scalable Geographic Routing Algorithms for Wireless Ad-Hoc Networks. *IEEE Network. Special Issue on Ad Hoc Networking: Data Communications and Topology Control*, 18(4):18–20, 2004.

[GA04] Anargyros Garyfalos and Kevin C. Almeroth. Coupon Based Incentive Systems and the Implications of Equilibrium Theory. In *IEEE International Conference on E-Commerce Technology. Proceedings*, pages 213–220. IEEE Computer Society, 2004.

[GFH05] Daniel Görgen, Hannes Frey, and Christian Hutter. Information Dissemination Based on the En-Passent Communication Pattern. In *Kommunikation in Verteilten Systemen (KiVS 2005)*, pages 129–141. Springer, 2005.

[GG03] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 31–42. USENIX, 2003.

[GHT05] Andreas Görlach, Andreas Heinemann, and Wesley W. Terpstra. Survey on Location Privacy in Pervasive Computing. In Philip Robinson, Harald Vogt, and Waleed Wagealla, editors, *Privacy, Security and Trust within the Context of Pervasive Computing*, The Kluwer International Series in Engineering and Computer Science, pages 23–34. Kluwer Academic Publishers, 2005.

[GHTM05] Andreas Görlach, Andreas Heinemann, Wesley W. Terpstra, and Max Mühlhäuser. *Handbook of Algorithms for Wireless Networking and Mobile Computing*, chapter Location Privacy, pages 393–411. Computer and Information Science Series. Chapman & Hall/CRC, 2005.

[GLAS99] J. J. Garcia-Luna-Aceves and Marcelo Spohn. Source-Tree Routing in Wireless Networks. In *Seventh Annual International Conference on Network Protocols, ICNP 1999*, pages 273–282. IEEE Computer Society, 1999.

[GLBML01] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for Sharing in Peer-to-Peer Networks. In *Electronic Commerce: Second International Workshop, (WELCOM)*, volume 2232 of *Lecture Notes in Computer Science*, pages 75–87, 2001.

[GRW05] Stefan Götz, Simon Rieche, and Klaus Wehrle. *Peer-to-Peer Systems and Applications*, volume 3485 of *Lecture Notes in Computer Science*, chapter Selected DHT Algorithms, pages 95–117. Springer, 2005.

[GSX02] Siddhartha K. Goel, Manish Singh, and Dongyan Xu. Efficient Peer-to-Peer Data Dissemination in Mobile Ad-Hoc Networks. In *International Conference on Parallel Processing Workshops*, pages 152–158. IEEE Computer Society, 2002.

[HCG+05] Pan Hui, Augustin Chaintreau, Richard Gass, James Scott, and Jon Crowcroft. Pocket Switched Networking: Challenges, Feasibility, and Implementation Issues. In *Autonomic Communication*, volume 3854 of *Lecture Notes in Computer Science*. Springer, 2005.

[HCS+05] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket Switched Networks and Human Mobility in Conference Environments. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251. ACM Press, 2005.

[HCW04] Elgan Huang, Jon Crowcroft, and Ian Wassell. Rethinking Incentives for Mobile Ad Hoc Networks. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 191–196. ACM Press, 2004.

[HDP03] Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha. Exploiting the synergy between peer-to-peer and mobile ad hoc networks. In *Proceedings of HotOS'03: 9th Workshop on Hot Topics in Operating Systems*, pages 37–42. USENIX, 2003.

[Hei06] Heise Zeitschriften Verlag. Marktforscher: Der weltweite Umsatz mit Handys schrumpft 2006 [online]. 2006. Available from: `http://www.heise.de/newsticker/meldung/68301` [seen 2007].

[Hei07] Andreas Heinemann. iClouds Project Site [online]. 2003-2007. Available from: `http://www.iclouds.tk.informtik.tu-darmstadt.de` [seen 2007].

[HFW99] Lars Erik Holmquist, Jennica Falk, and Joakim Wigström. Supporting Group Collaboration with Inter-Personal Awareness Devices. *Personal Technologies Journal*, 3(1–2):105–124, 1999.

[HKLM03a] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser. Ad Hoc Collaboration and Information Services Using Information Clouds. In Torsten Braun, Nada Golmie, and Jochen Schiller, editors, *Proceedings of the 3rd Workshop on Applications and Services in Wireless Networks, (ASWN 2003)*, pages 233–242, Bern, Switzerland, 2003. Institute of Computer Science and Applied Mathematics, University of Bern.

[HKLM03b] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser. iClouds – Peer-to-Peer Information Sharing in Mobile Environments. In Harald Kosch, László Böszörményi, and Hermann Hellwagner, editors, *Proceedings of the 9th International Euro-Par Conference, (Euro-Par 2003)*, volume 2790 of *Lecture Notes in Computer Science*, pages 1038–1045, Klagenfurt, Austria, 2003. Springer.

[HM05] Andreas Heinemann and Max Mühlhäuser. *Peer-to-Peer Systems and Applications*, volume 3485 of *Lecture Notes in Computer Science*, chapter Spontaneous Collaboration in Mobile Peer-to-Peer Networks, pages 419–433. Springer, 2005.

[HRS04] Andreas Heinemann, Johannes Ranke, and Tobias Straub. Zur rechtsverträglichen Technikgestaltung anhand einer M-Commerce-Anwendung. In *Mobile Economy – Transaktionen, Prozesse, Anwendungen und Dienste, Proceedings zum 4. Workshop Mobile Commerce*, volume P-42 of *LNI*, pages 162–177. Ges. für Informatik, 2004.

[HS03] Andreas Heinemann and Tobias Straub. Mund-zu-Mund-Propaganda mit Bonussystem in mobilen Ad-Hoc-Netzen. In *INFORMATIK 2003 – Innovative Informatikanwendungen, Band 2, Beiträge der 33. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, volume 35 of *LNI*, pages 366–371, Frankfurt am Main, Deutschland, 2003. Ges. für Informatik.

[HTA05a] Tobias Hoßfeld, Kurt Tutschku, and Frank-Uwe Andersen. Mapping of file-sharing onto mobile environments: Enhancement by UMTS. In *Mobile Peer-to-Peer Computing MP2P, in conjunction with the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'05)*, pages 43–54. IEEE Computer Society, 2005.

[HTA05b] Tobias Hoßfeld, Kurt Tutschku, and Frank-Uwe Andersen. Mapping of File-Sharing onto Mobile Environments: Feasibility and Performance of eDonkey with GPRS. In *Wireless Communications and Networking Conference, 2004. WCNC. 2005 IEEE*. IEEE Computer Society, 2005.

[HW05] Anna Hayes and David Wilson. Peer-to-Peer Information Sharing in a Mobile Ad Hoc Environment. In *Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, pages 154–162. IEEE Computer Society, 2005.

[IEE99] IEEE. IEEE 802.11 Wireless Specification, ISO/IEC 8802-11 [online]. 1999. Available from: `http://standards.ieee.org/getieee802/802.11.html` [seen 2007].

[Int05] International Telecommunication Union. The Internet Of Things [online]. 2005. Available from: `http://www.itu.int/internetofthings` [seen 2007].

[Int06] Intel Corporation. Haggle – pocket switched networking [online]. 2006. Available from: `www.haggleproject.org` [seen 2007].

[Iwa98] Yukari Iwatani. Love: Japanese Style [online]. 1998. Available from: `http://www.wired.com/news/culture/0,1284,12899,00.html` [seen 2007].

[JAP00] JAP Team. JAP Anon Proxy [online]. 2000. Available from: `http://anon.inf.tu-dresden.de` [seen 2007].

[JBRAS03] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards Realistic Mobility Models For Mobile Ad Hoc Networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 217–229. ACM Press, 2003.

[JLH$^+$99] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 195–206. ACM Press, 1999.

[JMB01] David B. Johnson, David A. Maltz, and Josh Broch. *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[JMC+01] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized Link State routing Protocol For Ad Hoc Networks. In *IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century*, pages 62–68. IEEE Computer Society, 2001.

[JMW05] Ari Juels, David Molnar, and David Wagner. Security and Privacy Issues in E-passports. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 74–88. IEEE Computer Society, 2005.

[Kaa03] Eija Kaasinen. User needs for location-aware mobile services. *Personal and Ubiquitous Computing*, 7(1):70–79, 2003.

[KBTR02] Abdelmajid Khelil, Christian Becker, Jing Tian, and Kurt Rothermel. An Epidemic Model for Information Diffusion in MANETs. In *MSWiM '02: Proceedings of the 5th ACM International workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, pages 54–60, New York, NY, USA, 2002. ACM Press.

[KH03] Jiejun Kong and Xiaoyan Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 291–302. ACM Press, 2003.

[KH06] Stan Kurkovsky and Karthik Harihar. Using Ubiquitous Computing in Interactive Mobile Marketing. *Personal and Ubiquitous Computing*, 10(4):227–240, 2006.

[KLW03] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks. In *IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, 2003.

[KLW04] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. Peer-to-Peer Computing in Mobile Ad Hoc Networks. In *Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures*, volume 2965 of *Lecture Notes in Computer Science*, pages 187–208. Springer, 2004.

[Kor02] Gerd Kortuem. *A Methodology and Software Platform for Building Wearable Communities*. PhD thesis, University of Oregon, 2002.

[KP04] Heiko Knospe and Hartmut Pohl. RFID Security. *Information Security Technical Report*, 9(4):39–50, 2004.

[KP06] Thabotharan Kathiravelu and Arnold Pears. What & When?: Distributing Content in Opportunistic Networks. In *2nd Int. Conference on Wireless and Mobile Communications (ICWMC'06)*, page 64. IEEE Computer Society, 2006.

[KSP+01] Gerd Kortuem, Jay Schneider, Dustin Preuitt, Thaddeus G. C. Thompson, Stephen Fickas, and Zary Segall. When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad hoc Networks. In *First International Conference on Peer-to-Peer Computing (P2P'01)*, pages 75–93. IEEE Computer Society, 2001.

[KST99] Gerd Kortuem, Zary Segall, and Thaddeus G. Cowan Thompson. Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles. In Hans-Werner Gellersen, editor, *Handheld and Ubiquitous Computing, First International Symposium, HUC'99, Karlsruhe, Germany, September 27-29, 1999, Proceedings*, volume 1707 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 1999.

[LM04a] Christopher Lueg and Omer Mahmood. Combining Mobile Data Transport and Mobile Data Recharging to Address Public Transport Information Maintenance Problems in Rural and Remote Australia. In *Information zwischen Kultur und Marktwirtschaft, Proceedings des 9. Internationalen Symposiums für Informationswissenschaft(ISI 2004)*, volume 42 of *Schriften zur Informationswissenschaft*, pages 337–348. Hochschulverband für Informationswissenschaft, 2004.

[LM04b] Christopher Lueg and Omer Mahmood. Mobile Data Transport Enabling Mobile Timetable Recharging in Rural and Remote Areas, UBICOMP'04 Poster Proceedings [online]. 2004. Available from: `http://ubicomp.org/ubicomp2004/adjunct/posters/lueg.pdf` [seen 2007].

[LV01] Arjen K. Lenstra and Eric R. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[LW02a] Christoph Lindemann and Oliver P. Waldhorst. A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. In *2nd International Conference on Peer-to-Peer Computing (P2P 2002), 5-7 September 2002, Linköping, Sweden*, pages 73–80. IEEE Computer Society, 2002.

[LW02b] Christoph Lindemann and Oliver P. Waldhorst. Eine Protokollumgebung für Peer-to-Peer Dokumentenaustausch in infrastrukturlosen mobilen Umgebungen. In *Mobile Ad-Hoc Netzwerke, 1. deutscher Workshop über Mobile Ad-Hoc Netzwerke WMAN 2002*, pages 167–178, 2002.

[LW05] Christoph Lindemann and Oliver P. Waldhorst. *Epidemic Data Dissemination for Mobile Peer-to-Peer Lookup Services*, volume 3485 of *Lecture Notes in Computer Science*, chapter 26, pages 435–455. Springer, 2005.

[LY02] Kalle Lyytinen and Youngjin Yoo. Issues and Challenges in Ubiquitous Computing. *Communications of the ACM*, 45(12):62–65, December 2002.

[Mas05] Massachusetts Institute of Technology. Reality Mining [online]. 2005. Available from: `http://reality.media.mit.edu/` [seen 2007].

[MdRK04] Ronald Mannak, Huib de Ridder, and David V. Keyson. The Human Side of Sharing in Peer-to-Peer Networks. In *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence*, pages 59–64. ACM Press, 2004.

[Neg06] Negev Software Industries, Ltd. NSIcom Website [online]. 2003-2006. Available from: `http://www.nsicom.com` [seen 2007].

[NM05] Martin Nilsson and Michael Mutschler. ID3v2 [online]. 1998-2005. Available from: `http://www.id3.org/` [seen 2007].

[NTT05] NTT DoCoMo, Inc. NTT DoCoMo Enhances Prototype Micro Fuel Cell for FOMA Handsets [online]. 2005. Available from: `http://www.nttdocomo.com/pr/2005/000676.html` [seen 2007].

[OKA+03] T. Ojala, J. Korhonen, M. Aittola, M. Ollila, T. Koivumki, J. Thtinen, and H. Karjaluoto. SmartRotuaari - Context-aware Mobile Multimedia Services. In *MUM 2003: 2nd International Conference on Mobile and Ubiquitous Multimedia*, pages 9–18, 2003.

[Ope06] Open Mobile Alliance. Wireless Application Protocol [online]. 1999-2006. Available from: `http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html` [seen 2007].

[OPT97] Donal O'Mahony, Michael Peirce, and Hitesh Tewari. *Electronic Payment Systems*. Artech House, 1997.

[PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244. ACM Press, 1994.

[PFH04] Alex Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking Connectivity in Developing Nations. *Computer*, 37(1):78–83, 2004.

[PN06] Kaustubh S. Phanse and Johan Nykvist. Opportunistic wireless access networks. In *AcessNets '06: Proceedings of the 1st international conference on Access networks*, pages 11–15. ACM Press, 2006.

[PPC06] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Comm.* , 44(11):134–141, 2006.

[PR99] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, USA, 1999. IEEE Computer Society.

[RC02] Anand Ranganathan and Roy H. Campbell. Advertising in a pervasive computing environment. In *WMC '02: Proceedings of the 2nd international workshop on Mobile commerce*, pages 10–14. ACM Press, 2002.

[RFJY03] Olga Ratsimor, Tim Finin, Anupam Joshi, and Yelena Yesha. eNcentive: A Framework for Intelligent Marketing in Mobile Peer-To-Peer Environments. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 87–94. ACM Press, 2003.

[RM03] Bharat Rao and Louis Minakakis. Evolution Of Mobile Location-Based Services. *Communications of the ACM*, 46(12):61–65, 2003.

[RSCH04] Åsa Rudström, Martin Svensson, Rickard Cöster, and Kristina Höök. MobiTip: Using Bluetooth as a Mediator of Social Context. In *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Adjunct Proceedings (demo)*, 2004.

[RT99] Elizabeth Royer and Chai-Keong Toh. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications*, 6(2):46–55, 1999.

[SB03] Christian Seitz and Michael Berger. Towards an Approach for Mobile Profile Based Distributed Clustering. In *International Conference on Parallel and Distributed Computing (Euro-Par 2003)*. Springer, 2003.

[SBB04a] Christian Seitz, Michael Berger, and Bernhard Bauer. MopiMine - Mobile Profile Mining. In *International Workshop on Wireless Ad hoc Networks (IWWAN '04)*, 2004.

[SBB04b] Christian Seitz, Michael Berger, and Bernhard Bauer. MPDG – Mobile Profile based Distributed Grouping. In *2nd IEEE International*

Conference on Pervasive Computing and Communications – Workshop on Mobile Peer-to-Peer Computing. IEEE Computer Society, 2004.

[SBB05] Christian Seitz, Michael Berger, and Bernhard Bauer. Towards a general approach to mobile profile based distributed grouping. *Personal Ubiquitous Computing*, 9(2):90–99, 2005.

[Sch97] Berry Schoenmakers. Security Aspects of the Ecash Payment System. In *State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography, Leuven, Belgium, June 3-6, 1997. Revised Lectures*, pages 338–352, 1997.

[Sch01] Rüdiger Schollmeier. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In *1st International Conference on Peer-to-Peer Computing (P2P 2001), 27-29 August 2001, Linköping, Sweden*, pages 101–102. IEEE Computer Society, 2001.

[SG02] Thomas Schwotzer and Kurt Geihs. Shark – a System for Management, Synchronization and Exchange of Knowledge in Mobile User Groups. *Journal of Universal Computer Science*, 8(6):644–651, 2002.

[SGG03] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia Systems Journal*, 9(2):170–184, 2003.

[SH04] Tobias Straub and Andreas Heinemann. An Anonymous Bonus Point System For Mobile Commerce Based On Word-Of-Mouth Recommendation. In Lorie M. Liebrock, editor, *Applied Computing 2004. Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 766–773, New York, NY, USA, 2004. ACM Press.

[SHCD06] James Scott, Pan Hui, Jon Crowcroft, and Christophe Diot. Haggle: A Networking Architecture Designed Around Mobile Users. In *IFIP WONS 2006*, 2006.

[SHR06] Tobias Straub, Manuel Hartl, and Markus Ruppert. Digitale Reisepässe in Deutschland - Prozesse und Sicherheitsinfrastruktur. In *Sicherheit 2006: Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 3. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v. (GI)*, pages 233–243, 2006.

[Sne01] Einar Snekkenes. Concepts for Personal Location Privacy Policies. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 48–57. ACM Press, 2001.

[SP02] Thomas Schwotzer and Thomas Preuss. Knowledge Exchange in Spontaneous Networks - Towards Ubiquitous Knowledge. In *E-World Syria - From Technology to E-Business (ET2EB 2002) Proceedings*, 2002.

[SS05] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, September 2005.

[Sta05] Thad E. Starner. Wearable Computing for the Developing World. *IEEE Pervasive Computing*, 4(3):87–91, 2005.

[Sun02] Sun Microsystems, Inc. Java Collections Framework [online]. 2002. Available from: `http://java.sun.com/j2se/1.5.0/docs/guide/collections/overview.html` [seen 2007].

[Sun06] Sun Microsystems, Inc. Java 2 Platform, Standard Edition (J2SE) [online]. 1994-2006. Available from: `http://java.sun.com/j2se/` [seen 2007].

[Swi03] Shockfish SA Switzerland. The SpotMe Homepage [online]. 2003. Available from: `http://www.spotme.ch` [seen 2007].

[THB$^+$02] Jing Tian, Joerg Haehner, Christian Becker, Illya Stepanov, and Kurt Rothermel. Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation. In *35th Annual Simulation Symposium*. IEEE Computer Society, 2002.

[Uni05] University of Bonn. BonnMotion - A Mobility Scenario Generation and Analysis Tool [online]. 2002-2005. Available from: `http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/index.html` [seen 2007].

[Var05] Various. MusicMoz [online]. 2001-2005. Available from: `http://musicmoz.org` [seen 2007].

[Var06] Various. Cryptix Project [online]. 1997–2006. Available from: `http://www.cryptix.org/` [seen 2007].

[VHM05] Marco Voss, Andreas Heinemann, and Max Mühlhäuser. A Privacy Preserving Reputation System for Mobile Information Dissemination Networks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 171–181. IEEE Computer Society, 2005.

[Wal05] Oliver P. Waldhorst. *Design and Quantitative Analysis of Protocols for Epidemic Information Dissemination in Mobile Ad Hoc Networks*. PhD thesis, Universität Dortmund, 2005.

[Wei91] Marc Weiser. The Computer of the 21st Century. *Scientific American*, 8(3):66–75, 1991.

[Wei93a] Marc Weiser. Hot Topics – Ubiquitous Computing. *IEEE Computer*, 26(10):71–72, 1993.

[Wei93b] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7):74–84, 1993.

[Wei94] Marc Weiser. The World Is Not A Desktop. *ACM Interactions*, 1(1):7–8, 1994.

[WGR05] Klaus Wehrle, Stefan Götz, and Simon Rieche. *Peer-to-Peer Systems and Applications*, volume 3485 of *Lecture Notes in Computer Science*, chapter Distributed Hash Tables, pages 79–93. Springer, 2005.

[WSRE03] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *First International Conference on Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2003.

[XOW04] Bo Xu, Aris Ouksel, and Ouri Wolfson. Opportunistic Resource Exchange in Inter-Vehicle Ad-Hoc Networks. In *2004 IEEE International Conference on Mobile Data Management (MDM'04)*, pages 4–12. IEEE Computer Society, 2004.

[Yah05] Yahoo! Inc. Yahoo! [online]. 2005. Available from: `http://www.yahoo.com/` [seen 2007].

[ZNS03] Gustavo Zurita, Miguel Nussbaum, and Mike Shaples. Encouraging Face-to-Face Collaborative Learning through the Use of Handheld Computers in the Classroom. *Lecture Notes in Computer Science*, 2795:193–208, 2003.

Appendix A

List of Abbreviations

The following symbols are used in Chapter 5:

P_a^-	entity *a*'s private key
P_a^+	entity *a*'s public key
$Cert_{P_a^+}$	a certificate for entity *a*'s public key
$S_{P_a^-}(msg)$	sign message *msg* with entity *a*'s private key
$V_{P_a^+}(msg)$	verify message *msg* with entity *a*'s public key
$C_{P_a^+}(msg)$	encrypt a message *msg* with entity *a*'s public key
$D_{P_a^-}(msg)$	decrypt a message *msg* with entity *a*'s private key

Erklärung[1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades "Dr. rer. nat." mit dem Titel "Collaboration in Opportunistic Networks" selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, 30. April 2007 Andreas Heinemann

[1] gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Wissenschaftlicher Werdegang des Verfassers[2]

1992 – 1999	Studium der Informatik, Nebenfach Mathematik Universität Tübingen
1999	Diplomarbeit *Fault Tolerance in Parallel Computing* in Kooperation mit der IBM Deutschland, Böblingen
30.11.1999	Abschluß des Studiums als Diplom-Informatiker
02/2002 – 01/2005	Mitglied im DFG-Graduiertenkolleg 492 “Infrastruktur für den elektronischen Markt” Technische Universität Darmstadt
WS 2002/2003	Lehrauftrag “Wirtschaftsinformatik 3” Fachhochschule Heilbronn
seit 02/2005	Wiss. Mitarbeiter im Projekt “SicAri - Eine Sicherheitsarchitektur und deren Werkzeuge für die ubiquitäre Internetnutzung” FB Informatik, Telekooperation, Technische Universität Darmstadt

[2] gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Publikationen von Andreas Heinemann

2006

[1] Jussi Kangasharju and Andreas Heinemann. Incentives for Electronic Coupon Systems. In *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*, pages 60–62. ACM Press, September 2006.

2005

[2] Andreas Görlach, Andreas Heinemann, and Wesley W. Terpstra. Survey on Location Privacy in Pervasive Computing. In Philip Robinson, Harald Vogt, and Waleed Wagealla, editors, *Privacy, Security and Trust within the Context of Pervasive Computing*, The Kluwer International Series in Engineering and Computer Science, pages 23–34. Kluwer Academic Publishers, 2005.

[3] Andreas Görlach, Andreas Heinemann, Wesley W. Terpstra, and Max Mühlhäuser. Location Privacy. In Azzedine Boukerche, editor, *Handbook of Algorithms for Wireless Networking and Mobile Computing*, Computer and Information Science Series, pages 393–411. Chapman & Hall/CRC, 2005.

[4] Andreas Heinamann and Max Mühlhäuser. *Spontaneous Collaboration in Mobile Peer-to-Peer Networks*, volume 3485 of *Lecture Notes in Computer Science*, chapter 25, pages 419–433. Springer, 2005.

[5] Marco Voss, Andreas Heinemann, and Max Mühlhäuser. A Privacy Preserving Reputation System for Mobile Information Dissemination Networks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 171–181. IEEE, 2005.

2004

[6] Andreas Heinemann, Max Mühlhäuser, Wesley W. Terpstra, and Erwin Aitenbichler. Allgegenwärtige Rechner: Eine neue Dimension der IT-Sicherheitsproblematik. *Thema Forschung 1/2004: IT-Sicherheit*, pages 100–105, March 2004.

[7] Andreas Heinemann, Johannes Ranke, and Tobias Straub. Zur rechtsverträglichen Technikgestaltung anhand einer M-Commerce-Anwendung. In Key Pousttchi and Klaus Turowski, editors, *Mobile Economy - Transaktionen, Prozesse, Anwendungen und Dienste, Proceedings zum 4. Workshop Mobile Commerce*, volume P-42 of *LNI*, pages 162–177, Augsburg, Deutschland, 2004. GI.

[8] Tobias Straub and Andreas Heinemann. An Anonymous Bonus Point System For Mobile Commerce Based On Word-Of-Mouth Recommendation. In Lorie M. Liebrock, editor, *Applied Computing 2004. Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 766–773, New York, NY, USA, 2004. ACM Press.

2003

[9] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser. Ad Hoc Collaboration and Information Services Using Information Clouds. In Torsten Braun, Nada Golmie, and Jochen Schiller, editors, *Proceedings of the 3rd Workshop on Applications and Services in Wireless Networks, (ASWN 2003)*, pages 233–242, Bern, Switzerland, 2003. Institute of Computer Science and Applied Mathematics, University of Bern.

[10] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser. iClouds – Peer-to-Peer Information Sharing in Mobile Environments. In Harald Kosch, László Böszörményi, and Hermann Hellwagner, editors, *Euro-Par 2003. Parallel Processing, 9th International Euro-Par Conference*, volume 2790 of *Lecture Notes in Computer Science*, pages 1038–1045, Klagenfurt, Austria, 2003. Springer.

[11] Andreas Heinemann and Tobias Straub. Mund-zu-Mund-Propaganda mit Bonussystem in mobilen Ad-Hoc-Netzen. In Klaus R. Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, and Wolfgang Wahlster, editors, *INFORMATIK 2003 - Innovative Informatikanwendungen, Band 2, Beiträge der 33. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, volume 35 of *LNI*, pages 366–371, Frankfurt am Main, Deutschland, 2003. GI.

2002

[12] Andreas Hartl, Erwin Aitenbichler, Gerhard Austaller, Andreas Heinemann, Tobias Limberger, Elmar Braun, and Max Mühlhäuser. Engineering Multimedia-Aware Personalized Ubiquitous Services. In *IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02)*, pages 344–351, December 2002.

[13] Max Mühlhäuser, Erwin Aitenbichler, Gerhard Austaller, Andreas Hartl, Andreas Heinemann, and Christoph Trompler. Towards Personalized Ubiquitous Computing Services. Technical Report TK-01/02, Fachbereich Informatik, TU Darmstadt, 2002.

2000

[14] Wolfgang Blochinger, Reinhard Bündgen, and Andreas Heinemann. Dependable High Performance Computing on a Parallel Sysplex Cluster. In Hamid R. Arabnia, editor, *Proc. of the Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000)*, volume 3, pages 1627–1633, Las Vegas, NV, U.S.A., June 2000. CSREA Press.