Enabling disruption tolerant services

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Dipl. inf. (Bioinformatik)

Jó Ágila Bitsch

aus Lagos, Portugal

Berichter:

Prof. Dr.-Ing. Klaus Wehrle
Prof. Dr.-Ing. Lars Wolf

Tag der Mündlichen Prüfung: 19.01.2017

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar

Reports on Communications and Distributed Systems

edited by
Prof. Dr.-Ing. Klaus Wehrle
Communication and Distributed Systems,
RWTH Aachen University

Volume 14

Jó Ágila Bitsch

Enabling disruption tolerant services

Shaker Verlag
Aachen 2017

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: D 82 (Diss. RWTH Aachen University, 2017)

Printed in Germany.

ISBN 978-3-8440-5164-3
ISSN 2191-0863

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

Disruption tolerant networking allows us to provide communication and information services in low connectivity and challenged network scenarios. These range from previously underserved rural areas to highly energy constrained special purpose sensor networks. Compared to wired-to-the-last-hop, infrastructure-based networks, infrastructure-less networks face a number of additional challenges, the fundamental one being varying connectivity over time. Building successful services for disruption tolerant networking requires three different key steps: (1) identification and characterization of possible applications and services, (2) optimization of neighbor discovery, and (3) efficient routing of data bundles.

We developed an initial model for users and services in the context of disruption tolerant networks that combines the service communication structure, with user experience and user expectations to predict a user's Quality of Experience. For this, we identify key characteristics of services and networks that make them suitable for challenged scenarios. These characteristics include: (1) Data Prefetching and Bundling, (2) Bursty Communication Style, (3) Soft Time Constraints, (4) Lenient Delivery Order Constraints, (5) One-Way Communication, and (6) Hop-to-Hop Usability. We propose initial quantitative metrics that may capture the user perceived quality. These metrics are: (1) Speed of Interaction, (2) Comfort and Frustration Level, (3) Delivery Effectiveness, and (4) Intercontact Times / Idle Period.

Overall, we argue that there is no particular *killer app* for DTNs, but rather, we show how considering these challenges to network communication in the design of an application can lead to a universally better experience for all users. In this context, we develop a set of sample applications particularly well-suited for challenging network environments, which are a collaborative event scheduling and a Wikipedia access application. We further explored map dissemination for indoor navigation and mobile sensor networks for wildlife monitoring.

We propose a new scheme for wireless neighbor discovery between unsynchronized nodes based on perfect difference sets. The construction is a mathematically provable optimal trade-off between the discovery delay and power consumption for discovering the presence of a potential communication partner in range. The application of Perfect Difference Sets (PDS) allows the provable optimal trade-off. Using PDS with a duty cycle of 2 % only has a discovery latency of about 4 min whereas U-Connect, the best related work, has a latency of 9 min, assuming a slot length of 100 ms. We further investigate discovery between devices with differing energy budgets. Through an exhaustive search of all possible combinations of schedule lengths and phase shifts, we show that PDS also performs fair in practice, that is with a duty cycle of greater or equal to 1 %.

To enable efficient routing of messages, we model human mobility in a novel way using only local information. We propose two routing protocols: SimBetAge which bases on the frequency that particular people met in the past, and GeoDTN, which works on the frequency at which certain locations were visited in the past. Instead of modeling the social interactions in a binary graph, in SimBetAge, we capture the temporal change of a social network in a weighted graph. Edges in the graph degrade over time, if not refreshed. In turn, we extend the definition of similarity and

betweenness to capture real-valued edge weights and further propose the directed ego flow betweenness as a novel metric to capture the usefulness of a node as a carrier for data bundles. We evaluated this approach against a number of existing routing schemes, in particular: Direct Delivery, Epidemic Routing, Prophet and the original SimBet. Using the MIT Reality Mining trace, the Haggle imote trace, and the Dartmouth Outdoor Experiment traces, we show that routing performance can be drastically improved when temporal changes are considered.

In GeoDTN, we modeled human mobility as a time dependent probability distribution around known anchor points. Based on this model, we developed a heuristic to predict future connectivity. To be able to evaluate this approach in the absence of widely deployed disruption tolerant consumer applications, we presented traces of self reported global mobility from an online social network website for 221 active users. The evaluation shows that geoDTN works similarly well as algorithms based on social group characteristics such as SimBetAge and outperforms binary movement algorithms as Move. In less periodic networks, geoDTN significantly outperformed the other algorithms in hop count and delivery time. It performs in general 130%, in individual cases even 200%, better in hop count. In terms of delivery time, it outperforms the related work on average by a factor of 3, PRoPHET by factor 6.

In combinations, these algorithms and methods can contribute to an overall disruption tolerant networking architecture, enabling improved services for all users, but in particular extending data network availability further into areas previously devoid of the otherwise necessary communication infrastructure.

Kurzfassung

Unterbrechungstolerante Kommunikation (Delay Tolerant Networking, DTN) erlaubt die Datenübertragung in Netzwerken mit geringer Konnektivität. Hierzu gehören beispielsweise bisher diesbezüglich unterversorgte ländliche Gebiete als auch sehr energiebeschränkte Sensornetze. Im Gegensatz zu vollständig verkabelten, infrastruktur-basierten Netzwerken bringen solche infrastrukturlosen Sensornetze eine Reihe von zusätzlichen Problemen mit sich, wobei die sich über die Zeit ändernde Konnektivität die größte Herausforderung darstellt. Für die Entwicklung von unterbrechungstoleranten Netzen und diese Netze nutzende Dienste bedarf es also dreier grundlegender Schritte: (1) die Identifikation sinnvoller Anwendungen und Dienste sowie deren Beschreibung, (2) die effiziente Erkennung von potentiellen Kommunikationspartnern in Funkreichweite und (3) die effiziente Vermittlung von Nachrichten.

Im Rahmen dieser Arbeit wurde ein initiales Nutzer- und Dienstmodell zur Verwendung in unterbrechungstoleranten Netzen entwickelt. Dieses Modell kombiniert Aspekte der Kommunikationsstruktur eines Dienstes mit dem Nutzerkontext, der aus bisherigen Erfahrungen und Erwartungen des Nutzers an einen Dienst aus besteht. Basierend darauf kann letztlich dessen Zufriedenheit, die „Quality of Experience", vorausgesagt werden. Hierfür wurden die Kerncharakteristika identifiziert, welche Dienste und Netzen für diese Art von Kommunikation besonders geeignet machen. Die sind (1) „Data Prefetching" und „Bundling", (2) stoßweise, diskontinuierliche Kommunikation, (3) weiche Zeitschranken, (4) weiche Schranken zur Auslieferungsreihenfolge, (5) Einwegkommunikation und (6) Nützlichkeit von Daten auf jedem Hop. Ergänzend schlagen wir folgende initialen quantitativen Metriken zur Bewertung der Nutzerzufriedenheit vor: (1) die Interaktionsgeschwindigkeit, (2) Komfort- und Frustrationslevel, (3) die Effektivität einer eingegangenen Nachricht im Auslösen von Nutzerreaktionen, sowie (4) die Dauer zwischen technischen Kommunikationsgelegenheiten.

Insgesamt gibt es keine spezielle Killeranwendung für DTNs. Eine bessere Berücksichtigung der begrenzten Netzwerkressourcen kann aber universell zu einer besseren Nutzererfahrung führen. In diesem Kontext wurden eine Reihe von Beispielanwendungen entwickelt, die besonders für DTNs geeignet sind. Hierzu gehören eine kollaborative Terminfindung, der Zugriff auf Wikipedia, die Verteilung von Kartenmaterial zur Gebäudenavigation sowie mobile Sensornetze zur Tierbeobachtung.

Weiterhin wurde ein neues Schema (PDS) zur drahtlosen Erkennung zeitlich nicht synchronisierter Netzknoten in Funkreichweite vorgeschlagen, das auf Differenzenmengen basiert. Mathematisch beweisbar erzeugt diese Konstruktion den optimalen Kompromiss zwischen Erkennungslatenz und Energieverbrauch. PDS erreicht bei einer angenommen Funkbakenrate von 10 Hz und einem Tastgrad von 2% eine maximale Latenz von 4 min, während U-Connect, die bisher beste verwandte Lösung, unter gleichen Bedingungen auf eine maximale Latenz von 9 min kommt. Weiterhin wurde die Erkennungsrate von Netzknoten mit unterschiedlichen Energiebudgets untersucht. Durch eine vollständige Suche aller Kombinationen von Zykluslängen und Phasenverschiebungen konnte gezeigt werden, dass PDS sich in der Praxis bei allen Tastgraden größer 1% fair verhält.

Zur effizienten Weiterleitung von Nachrichten wurden menschliche Bewegungsmuster auf eine neue Art modelliert, welche nur lokale Daten in einem Netzknoten benötigt. Hierbei werden zwei neue Routingprotokolle vorgeschlagen: SimBetAge basiert auf der bisherigen Häufigkeit des Treffens von Personenpaaren während GeoDTN auf den bisherigen geographischen Aufenthaltswahrscheinlichkeiten von Personen basiert. So wird in SimBetAge eine soziale Beziehung nicht als binärer, sondern als gewichteter Graph modelliert, was es erlaubt, zeitliche Veränderungen zu erfassen. Entsprechend werden die Definitionen von Ähnlichkeit und Intermediationszentralität auf reellwertig-gewichtete Graphen erweitert. Außerdem wird eine Metrik der gerichteten Egozentralität vorgeschlagen, welche die Eignung eines Netzknotens zur Weiterleitung von Nachrichten abbildet. Der Ansatz wurde mit einer Reihe von Routingsalgorithmen verglichen, insbesondere mit „Direct Delivery“, epidemisches Fluten, PRoPHET und dem originalen SimBet. Unter Verwendung der Datensätze „Reality Mining“, „Haggle iMote“ und „Dartmouth Outdoor Experiment“ wurde gezeigt, dass die Berücksichtigung der zeitlichen Veränderlichkeit von sozialen Beziehungen die Performanz drastisch verbessern kann.

In GeoDTN wird menschliche Mobilität als eine zeitabhängige Aufenthaltswahrscheinlichkeit um Ankerpunkte modelliert. Dies erlaubt die Entwicklung einer Heuristik zur Vorhersage der zukünftigen Konnektivität. Da bisher noch keine weitverbreiteten DTN Anwendungen existieren, wurde als Datenbasis zur Bewertung des Ansatzes die selbstdeklarierte Mobilität von 221 sehr aktiven Nutzern aus einem sozialen Online-Netzwerk verwendet. Die Evaluation zeigt, dass geoDTN eine ähnliche Performanz aufweist wie Algorithmen auf der Basis sozialer Charakteristika, wie etwa SimBetAge, und Algorithmen wie Move deutlich übertrifft. In Netzwerken mit weniger periodischer Mobilität übertrifft geoDTN andere Algorithmen in seiner Performanz deutlich in Bezug auf Hop-Anzahl; im Durchschnitt um 130% und in Einzelfällen um bis zu 200%. Bezüglich der Zustellungsgeschwindigkeit ist geoDTN im Schnitt um einen Faktor 3, gegenüber PRoPHET sogar um den Faktor 6, performanter.

In Kombination tragen diese Algorithmen und Methoden zu einer übergeordneten DTN Architektur dabei, erlauben verbesserte Dienste für Nutzer und verbessern den Zugang zu Datennetzen insbesondere in Gebiete, in denen die sonst nötige Kommunikationsinfrastruktur fehlt.

Acknowledgments

For Roann.

For Migi.

For Jami.

For Angel Bear.

For Mama.

For Oma.

For Maomé.

For my family.

For my students.

For my colleagues.

For my friends.

Finally!

Thank you!

Contents

1

Introduction

Disruption tolerant networking allows us to provide communication and information services in challenging scenarios. These range from previously underserved rural areas to highly energy constrained special purpose sensor networks.

Compared to wired-to-the-last-hop, infrastructure-based networks, infrastructure-less networks face a number of additional challenges. The fundamental one is *varying connectivity over time*. Kevin Fall [Fal03] identifies three main challenges leading to this variation: (1) path and link characteristics, (2) network architectures, and (3) end system characteristics.

The constraints resulting from the path and link characteristics are an inherent physical property of the scenario. Communication between two nodes may only be available intermittently, because of node mobility or environmental changes. Signal propagation and bandwidth under water or over very long distances are limited.

Similarly, the network architecture, also *network layout* or *setup*, is limited by economical and organizational constraints. Interoperability with co-located services and nodes or existing legacy applications can be problematic to realize. Security goals such as service availability and access control are fundamentally more challenging than in a wired network with continuous end to end connectivity.

In contrast, the design space for end systems is more open. While they typically have only limited resources available and therefore have to work under a low duty cycle to achieve a required longevity, the physical and in particular also the software components are much more malleable.

Given these general characteristics, we focus on these end systems to enable usable and efficient disruption tolerant networks. First, we analyze the core requirements and consequences for communication in challenging environments. We look at connectivity traces and existing applications and derive different classes of services and users. This understanding helps us to create a node discovery mechanism that fulfills the energy and timing requirements necessary for successful service delivery. Using the insights gained, we develop two routing schemes specifically for social networks based on human mobility. Finally, we present a set of possible sample applications benefiting from this process.

In the following, we discuss the specific research questions arising from this problem areas, including different application and user models, efficient neighbor discovery, and routing of data bundles.

1.1 Problem Statement

Building successful services for disruption tolerant networking requires three different key steps: (1) identification and characterization of possible applications and services, (2) optimization of neighbor discovery, and (3) efficient routing of data bundles.

- **Identification and characterization of suitable services:** Services and applications need to be characterized in terms of communication requirements, interaction patterns and chattiness. Similarly, users and user expectations need to be modeled and evaluated against satisfiability with respect to physical, technical, and conceptual constraints.
- **Energy-efficient neighbor discovery with minimal latency:** As neighbor discovery dominates the overall energy budget of a mobile node, it is adamant, that we minimize the power consumption of a neighbor discovery scheme given a specific latency requirement.
- **Routing with minimal delay and resource overhead:** At their core, routing schemes in disruption tolerant networks function as an oracle of future connectivity. While previous schemes exist, they typically ignore the change of social and geographic contact graphs over time, thereby degrading over longer periods of time. Being aware of these changes and taking them into account potentially leads to better routing decisions.

Each of these major problem areas raises specific research questions. We identify general observations that underlay our approach and then list our contributions towards usable and efficient services in disruption tolerant networks.

1.2 Identification and Characterization of Services

To better serve the user in a disruption prone communication setting, we need to carefully consider the different aspects which affect the user's experience. Failure to meet a user's expectations will lead to the service not being accepted on the market.

As networks in challenging environments are inherently constrained, we need to first identify the limitations of such scenarios. This is not restricted to network characteristics, but also includes service requirements and the background of the user. The subjective nature of user experience as an integral part of service acceptance, leads us to the understanding that traditional Quality of Service (QoS) metrics are insufficient. Therefore, we need to introduce a framework for understanding *Quality of Experience (QoE)* in challenged networks. Such a framework needs to answer the following research questions:

- **How to characterize the network environment?** This includes physical nodes, the network topology, link availability and efficiency, and mobility.
- **How can a user's expectations be modeled?** Different user groups may have diverging expectations and backgrounds, that need to be taken into account.
- **How can services be characterized?** Different service classes have different chattiness and have different communication requirements in terms of bandwidth and delay tolerance.
- **How do these characteristics influence user acceptance?** Services have widely differing requirements, leading to different user acceptance within the same network environment.

1.2.1 Observation: There is no singular Killer App for DTNs

In terms of evaluation and usability, today's communication protocols base their measure of usability on throughput, hop count, and delivery time. While these numerical measures allow us to capture certain aspects of the performance, they are only of limited use in networks in challenged environments. The conditions of the network itself are the dominant factors. We thus require new measures based on softer, more qualitative factors to properly capture the utility for a user of emerging protocols for challenged networks.

While the community is on a quest for *the* DTN killer app [LH09], we argue that most if not all mobile applications would highly benefit from becoming aware of delays and disruptions, and varying connectivity over time.

1.2.2 Contribution: User and Service Models for DTNs

To answer the research questions in previously outlined in Section 1.2, we develop a model for users and services in the context of disruption tolerant networks. As previously argued, the network structure and with it commonly used Quality of Service metrics are insufficient to evaluate service acceptance by its users. The service communication structure, together with user experience and user expectations factor into the resulting outcome for the user, the *Quality of Experience*.

Accordingly, we identify characteristics of services and networks that are suitable for challenged scenarios. We propose new metrics that allow the Quality of Experience evaluation of services and give guidelines on how to evaluate them, based on those results.

Finally, we develop a *set of sample applications* particularly well-suited for challenging network environments. Through this, we show how considering these challenges to network communication in the design of an application can lead to a universally better experience for all users.

1.3 Neighbor Discovery with Minimal Latency

As we will explore in Section 2.4, communication opportunities in challenged networks are rare and short. Therefore, the task to detect such an communication opportunity needs to run continuously. This has two conflicting consequences for the employed *neighbor discovery* algorithm. (1) It needs to consume as little energy as possible, and (2) the delay from the point in time such an opportunity arises to the detection of it needs to be as short as possible. As nodes may neither be synchronized, nor have symmetrical energy requirements, we need to find a scheme that is effective and fair given the constraints of the scenario. Therefore, such a neighbor discovery framework needs to answer the following research questions:

- **How to discover communication opportunities where potential communication partners are not synchronized?** As user devices are not accurately synchronized to the same clock, a discovery mechanism must deal with arbitrary clock offsets.
- **How to ensure minimal discovery latency with minimum energy use?** The acceptable discovery latency is highly dependent on the use case. However, for a given acceptable latency, energy expenditure should be minimal.
- **How to ensure discovery when nodes have different energy budgets?** Depending on their function in a network, differing battery capacity or energy harvesting intake, nodes may have different energy budgets available for neighbor discovery.
- **What is fairness in neighbor discovery?** Typical metrics for neighbor discovery include the latency for a given energy budget and the energy requirements given a specific latency bound. For the asymmetric case, no useful overall metrics exist yet.

1.3.1 Observation: Cyclic Block Designs Fit Requirements

Block design in combinatorics [CD06] is typically used for experimental design. It represents a family of related subsets of a set of elements that fulfill specific properties. In particular, perfect difference sets, where each subset of a specific block design has exactly one element in common with all other subsets of the block design, fit the problem of neighbor discovery: For each pair of communication nodes, the radio activation schedule needs to guarantee that they are active at the same time at least once during a discovery period.

In particular, perfect difference sets are provably the smallest sets of radio activation schedules that fulfill all neighbor discovery requirements.

1.3.2 Contribution: Perfect Difference Sets for Discovery

We propose and evaluate a new scheme for wireless neighbor discovery between unsynchronized nodes based on perfect difference sets. The solution presents a construction for a mathematically provable optimal trade-off between the discovery delay and power consumption making use of an exact formulation of the problem in set theory. Dividing time into periodical time slots, a neighbor discovery schedule has to fulfill a number of properties: (1) The intersection between different schedules cannot be empty. (2) The schedules should not be trivial, as in the radio is always on. (3) All schedules should have equal size. (4) The schedule should be closed under rotation. The last property ensures that two schedules intersect for all possible phase shifts, therefore allowing unsynchronized devices to discover each other.

The application of Perfect Difference Sets (PDS) allows the provable optimal trade-off. Further, we generalize from property 3, to allow the discovery of devices with differing energy restrictions.

We present a proof of concept implementation of this scheme for the Linux kernel as well as for TinyOS based sensor nodes, which are the basis for further energy efficient local content sharing applications. In this context, we also elaborate on the design of active radio slots, and how applications can announce new and changed content in an energy efficient way.

1.4 Routing with minimal delay and overhead

We do not only want to forward data packets—often called *bundles* in the context of disruption tolerant networks—in the local broadcast domain, but to a possibly far away node. Under the constraint of limited storage, where we cannot simply forward all messages to everyone, we need to identify which communication partner is the better carrier for a bundle. As we do not have global information available in the general case, we need to define criteria, that can be evaluated locally and are still effective in *routing* a bundle towards its destination. Moreover, we need to consider the evolution of connectivity patterns over time. This relates to social patterns as well as geographical pattern.

Therefore, such routing schemes need to answer the following research questions:

- **How to predict future connectivity between nodes in human driven mobility?** To know which node is the best carrier of a message, we need to know, which node will get the message closest to its destination.
- **What locally available information helps with predictions?** For a practical routing scheme, only information that is available locally can be evaluated. This includes a local node's historical behavior or relatively static global information, such as geographical maps.
- **How to ensure fairness between forwarding nodes?** With only limited resources available at a node, the more *central* nodes may be over-used. Can a routing scheme be extended to balance the load at the cost of slightly longer delivery latency?

1.4.1 Observation: People Meet People They Met Before

The assumed mobility patterns for most Mobile Ad-hoc Network (MANET) protocols are various forms of random walks, ignoring social and geographical aspects of human mobility. Human mobility incorporates a number of characteristics: Friendship, familiarity and employment aspects guide who we meet how regularly. Humans are territorial. The likelihood of coming back to a place, we visited in the past is much higher than the likelihood of returning to a place we never visited. Routing schemes taking these characteristics into account may have the potential to significantly improve routing performance.

1.4.2 Contribution: Human Mobility Based Forwarding

We are proposing two novel approaches for predicting human mobility using local information for forwarding messages in disruption tolerant networks: (1) Social Network Based Forwarding, which takes into account, who met whom how often in the past, and (2) Geography Based Forwarding, which works on the frequency at which certain locations were visited in the past.

1.4.2.1 Social Network Based Forwarding

While the social structure has been identified as a useful heuristic for routing in challenged networks, the gradual change over time in the social structure has been neglected. While definitions of similarity and betweenness for challenged networks [DH07] may be useful at a short time span, its performance degrades over time, as eventually all devices have met each other at least once.

Instead of modeling the social interactions in a binary graph, a weighted graph where the weights between nodes degrade over time, if they are not refreshed, allows to capture this temporal change. We extend the definition of similarity and betweenness to capture real-valued edge weights and further propose the "directed ego flow betweenness" as a novel metric to capture the usefulness of a node as a carrier for data bundles.

1.4.2.2 Geography Based Forwarding

While human social interaction is one key heuristic in understanding human mobility, the actual locations visited in the past are also a good predictor on where users will be in the future. Accordingly, we model human mobility as a probability distribution around known anchor points. This geographic probability function changes over time, just as social interactions do. Nevertheless, in a similar process to the modeling of social ties, we can model the probability of locations as degrading over time if not refreshed.

Based on this model, we develop a heuristic to predict future connectivity to be used as a connection oracle for disruption tolerant networking. In particular, this addresses regional and global mobility.

To be able to evaluate this approach in the absence of widely deployed disruption tolerant consumer applications, we present traces of self-reported global mobility. Using the advent of location based online social networks allowed us to collect global location data for a variety of users.

1.5 Structure

The remainder of this work is structured as follows: Chapter 2 discusses the general context and concepts fundamental to this work. We present the idea behind disruption tolerant networks and common use cases, as well as methods for deriving global mobility traces. In Chapter 3, we discuss our first major contribution: the identification and characterization of suitable services. This is followed by an in-depth description and analysis of our second major contribution: energy-efficient neighbor discovery with minimal latency in Chapter 4. Chapters 5 and 6 discuss the third major contribution: Routing with minimal delay and resource overhead basing on the social network of carriers and their geographical timeline respectively. Chapter 7 presents applications of DTNs that we developed over the course of this work. Finally, we conclude this work in Chapter 8.

2

Background

Disruption and delay tolerant networks are a subclass of mobile networks enabling communication in environments without continuous end to end connectivity. In this chapter, we introduce the underlying concepts behind these networks and discuss use cases and common requirements.

2.1 Challenges to Setting Up Mobile Networks

Besides voice calls and text messaging, mobile phone network providers can offer mobile internet connectivity to customers with download speeds up to 300 Mbit/s for LTE. Still, typical download speeds in urban areas are 11 Mbit/s to 23 Mbit/s in Germany[1]. However, as soon as a user leaves urban areas, connectivity and bandwidth drops significantly.

While the cost of user equipment, typically referred to as client or mobile station, is borne by end users, wireless network access requires a significant investment into infrastructure by the mobile network carrier [HT06, p.153].

Even with the core network in place, the base stations, used to connect mobile terminals, require significant financial and operational investment.

- **Connectivity:** Depending on the location of a base station, the provider needs to connect the base station to its core network. This might be through physical cables, e.g., DSL lines, or dedicated microwave links.
- **Power:** Base stations use significantly more power than mobile terminals. This requires either mains power or power generation, e.g., solar panels or wind turbines.

[1] `http://www.netindex.com/mdownload/2,27/Germany/`

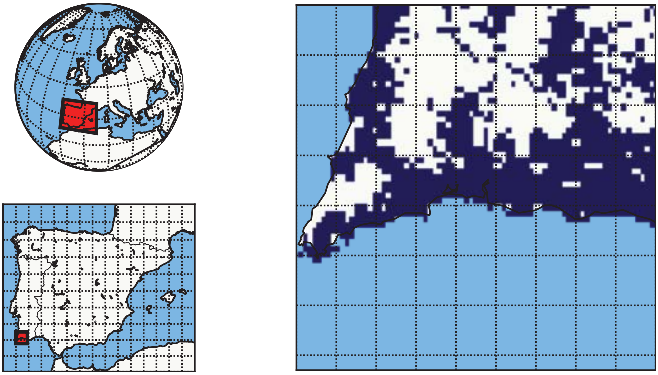


Figure 2.1 3G Coverage in South West Portugal. While population centers are well covered, rural less touristic areas only have limited Internet connectivity.

- **Physical Structure:** Radio propagation depends highly on the local topography. As a general rule, the higher an antenna, the wider the range. In addition, the base station may need to be protected from vandalism and theft.

Each cell only provides limited range up to a maximum of 35 km [Zha12, p. 8]. As most technologies provide roaming and hand-over, users can move seamlessly between adjacent cells. However, while urban centers are well covered, it is often not economically viable for network providers to install base stations outside population centers.

To illustrate this, Figure 2.1 shows current 3G network coverage[2] in South West Portugal, the author's region of birth. We see that as population centers are well covered, the rural, less touristic areas have indeed only limited Internet connectivity. To properly serve these areas, we need to explore complementary approaches [EB15]. Summarized under the umbrella term *challenged networks*, these types of networks make data networking available in low infrastructure regions.

2.2 Challenged Networks as a Solution Approach

The term DTN, short for *Delay Tolerant Network* or *Disruption Tolerant Network* describes a class of *challenged networks*, where nodes do not have continuous end to end connectivity, either due to physical limitations or due to unpredictable mobility. They are characterized by:

- **No end to end path:** DTNs typically do not rely on stable end to end paths between all nodes of the network. Growing network complexity and a rising number of participating nodes exacerbates this problem.

[2] `http://maps.mobileworldlive.com/network_info.php?nid=726&org_id=3999&cid=31`

- **Unreliable links:** Even when a direct link between two nodes exists, it may drop at any time, e.g., because of node mobility. In addition, a link may be asymmetric.
- **Sparse connectivity:** Depending on the topology and mobility of network nodes, they might not be able to communicate over extended periods of time.
- **Gateway nodes:** A network typically has at least one gateway node providing stable connectivity to the Internet.

In terrestrial use cases—the main focus of this thesis—DTNs must deal with the following challenges:

- **Mobility:** The nodes that form such a network are typically mobile. This mobility can be *self-controlled*, as is the case for robotic vehicles, or *passively controlled*, as is the case for human or animal carried mobile devices.
- **Size and Weight Constraints:** As a direct consequence of the mobility, especially in the passive case, the physical size and weight of the mobile device is constrained. This limits the size and thereby the capacity of batteries, but also the size of antennas.
- **Intermittent Energy:** Following external limits on the weight of a node, the capacity of used batteries, and thereby the energy supply is limited. While the node's mobility allows for recharging cycles, these are intermittent and the energy balance needs consideration in the design of DTN systems.

These fundamental challenges inform the design decisions of the rest of this work. In the following sections, we discuss the fundamental approaches of delay and disruption tolerant networks and commonly accepted use cases.

2.3 Delay and Disruption Tolerant Networks

To address the challenges put forth in the previous section, DTNs use a *store-carry-forward* approach as depicted in Figure 2.2. When timely direct delivery of a message from a source to a destination is not possible, the source node will forward the message to an intermediary, which *stores* the message for the time being. Afterwards, this intermediary physically moves to a different location—it *carries* the message. Finally, when the intermediary is in range of the destination or a node better suited for delivery to the destination, it *forwards* the message.

The terminology distinguishes two related but conceptually different types of DTNs: *Delay* Tolerant Networks and *Disruption* Tolerant Networks. *Delay* Tolerant Networks are characterized by long but predictable delays. Causes for these delays can be (1) the speed of light, e.g., for communication between Earth and Mars, or (2) physical occlusion, e.g., a satellite around Mars will not have continuous connectivity to a Mars rover, as the planet will be physically in the way. However, we can predict

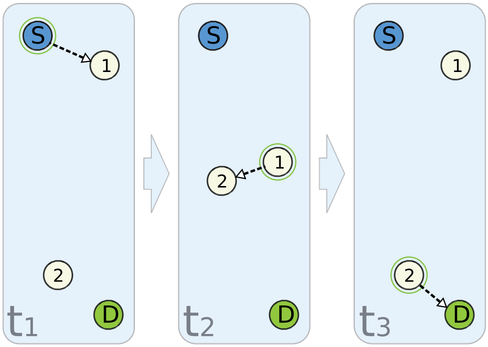


Figure 2.2 Example of a store-carry-forward bundle delivery. Node S sends a message for node D but no opportunity for direct delivery exists. However, node 1 and 2 may be used as relaying nodes. At time t_1, S forwards the message to node 1, at time t_2 node 1 forwards the message to node 2 and, finally, at time t_3 node 2 delivers the message to the destination D.

connectivity accurately from the flight path of the satellite and the location of the rover.

In contrast, *Disruption* Tolerant Networks are characterized by less predictable disruptions and movement patterns. As nodes are often carried by humans in these kinds of networks, they are also referred to by the name *Pocket Switched Networks* [HCG+06]. In the following, we discuss a range of DTN use cases.

2.3.1 DTN Use Cases

The working group DTN within the IETF, an international standardization body for internet protocols formed in 2014 following the earlier developments within the DTN research group (which formed in 2002 and dissolved in 2016) within the IRTF, a research focused sister organization of the IETF. The goal of the WG DTN is the standardization of exchange formats and protocols for DTNs. As a first step, the group collected use cases for these protocols [Mar15]. In the following, we present these use cases and their particular requirements, which include space and interplanetary, as well as terrestrial applications.

2.3.1.1 Space Systems Communications

This use case focuses on the use of DTN concepts for space systems. We distinguish between near earth and deep space communications. Near earth communication includes the communication with the International Space Station (ISS) and using Tracking and Data Relay Satellite (TDRS). Using DTNs instead of classic communication paradigms has the potential to improve availability and make better use of the bandwidth, while dealing with varying latency.

Satellites in geosynchronous orbit face relatively high delay and frequent disruptions, while low earth orbit satellites move out of range of a particular ground station relatively fast. DTN's potential lies in the better handling of scheduled intermittent contacts but also higher resilience to disruption.

Similarly, deep space communication, i.e., communication with space probes such as *New Horizons*, *Cassini* and various Mars missions, face a number of communication challenges, that could benefit from DTNs, as standard Internet protocols such as TCP fail [WPPSM07] at these distances. The reason for this lies in the physical distance between Earth and the space probe. The distance to Mars ranges from 4 to 21 light minutes and TCP timeouts, such as the RTO having a maximum value of 64 seconds. Even with increased RTO, performance degrades significantly under packet loss.

Also, isolated ground systems, such as Mars rovers or the Philae lander launched from the Rosetta space craft exploring the comet Tschuri, can only exchange data during satellite over-flights.

Specific Requirements

- **Limited resources:** Space crafts are severely resource-constrained. Communication protocols should incur minimal data overhead, as the data rate of the communication channel is very low. Similarly, energy and processing resources are limited on the spacecraft.
- **Integrity and Security:** From an operational perspective, the DTN WG identifies end-to-end integrity assurance and streamlined security management as specific requirements.

2.3.1.2 Unmanned Air Systems in Integrated Air Space

As unmanned air systems start to operate in the same airspace as civil aircrafts, the DTN WG identifies telemetry and control of these vehicles via DTN protocols, as a promising direction to deal with occasional disruptions of connectivity. The group claims that Internet protocols, such as TCP/IP, fail to provide adequate safety of flight, especially in remote regions where terrain features and atmospheric conditions may lead to extended outages.

Specific Requirements

- **Security:** The protocol must ensure only authorized entities can control the system. This has implications for disruption tolerant security key management. Depending on the application, telemetry information may be confidential.
- **Dynamic routing:** Moving in and out of coverage areas rapidly requires that the network paths and be changed quickly and dynamically.
- **Efficient neighbor discovery:** As a direct consequence, possible neighbors and contacts need to be identified quickly and energy efficiently.
- **Tight, adjustable control loops:** The control software must be aware of the network conditions and react appropriately, e.g., by slowing down the air system or immediately leaving potential hazardous areas.

2.3.1.3 Unmanned Underwater Vehicles (UUV)

Like unmanned air systems, unmanned underwater vehicle networking can greatly benefit from improved DTN performance. While the technology is still in its exploratory phase, underwater communication is sound based. The comparatively low speed of sound underwater of about 1.5 km/s and the high frequency dependent attenuation leads to low data rates at comparatively short distances. At the same, research aims at operations that might take up to several months. Communication within an UUV may either be scheduled or based on opportunistic contacts. Like over-land applications, the DTN working group expects the mobile UUVs to form delay tolerant multi-hop networks.

Specific Requirements

- **Dynamic routing:** This use case requires dynamic routing schemes running on resource and bandwidth constrained devices.
- **Efficient neighbor discovery:** To enable this, possible neighbors and contacts need to be identified quickly and energy efficiently.
- **Streamlined security:** According to the group, streamlined security would improve operations and set-up.

2.3.1.4 Civil Aviation

The communication networks used for communication between the Air Traffic Control (ATC) and the Airline Operations Control (AOC) themselves—the global Aeronautical Telecommunications Network (ATN)— uses of traditional non-DTN protocols. However, it is optimized for low delay and high reliability.

As the status of the aircraft is tracked over all phases of a flight though, different management and maintenance information may experience notable delays and disruption. Specific data links can be disrupted at any time or may not be available during take-off and landing or over the ocean. In this context, the use of DTNs may be helpful, to ensure information transfer.

Specific Requirements

- **End-to-end integrity:** For the safety of the aircraft, it is important that status and maintenance information is protected against tampering.
- **Security Key Management:** Similarly, only authorized parties should have access to the information.

2.3.1.5 Vehicular Delay Tolerant Networks

Vehicular networks are useful for communicating road conditions, traffic monitoring, road safety, and navigation. However, contact opportunities are short and have only limited bandwidth. Factors such as sparse traffic and terrain features further limit communication opportunities.

Specific Requirements

- **Security Key Management:** To ensure the reliability of safety critical information, key management for reputation and security management is necessary.
- **Dynamic Routing:** While mobility may be limited to streets, different routes and speeds, lead to dynamic mobility. Routing needs to consider these factors.
- **Neighbor and Contact Discovery:** Possible neighbors and contacts need to be identified quickly to maximize the available data transfer volume.

2.3.1.6 Disaster Response and Humanitarian Aid

After a natural and man-made disasters such as tsunamis or terrorist attacks, communication infrastructure can be severely damaged and become unavailable completely. To ensure coordination of the emergency response teams, these teams typically use portable or vehicular radio systems with a terrestrial or satellite backhaul. Depending on the nature of the disaster, the outage may extend over longer periods of time and nodes may be lost to wear and tear, battery life time.

Specific Requirements

- **Security Key Management:** Safety critical information must be reliable and may be confidential. Key management and security management must be tuned this use case.
- **Network Management:** Different node types, such as building stability sensor networks, but also personal communication devices and relays need appropriate management to ensure the best possible operation.
- **Neighbor and Contact Discovery:** Possible neighbors and contacts need to be identified quickly to maximize the available data transfer volume.

2.3.2 Contact opportunity

Communication consists of two phases: the discovery phase and the communication phase. As a node spends most of its time in the discovery phase, it is of utmost importance that this process is energy efficient. We will focus on this phase in Chapter 4. Once discovery is complete, the node enters the communication phase. In this phase, energy efficiency is less important and throughput and routing become the dominant goal. The beginning of the communication phase—the *handshake*—is structured as follows:

1. **Successful neighbor discovery**: A potential communication partner entered physical communication range. Communication becomes physically possible.
2. **Data freshness**: Ensure data changed since the last encounter. New data is potentially available for transfer and it makes sense to continue this handshake.

3. **Exchange of routing information**: Exchange metrics such as contact statistics, location statistics and energy statistic. This step is highly dependent on the routing scheme.

4. **Message carrier selection and prioritization**: Decide which node is the best carrier for the message and sort them by utility. This depends on the routing scheme, recipient, size and deadline of a message. The most important messages will be transmitted first, to ensure the highest probability of successful transfer before the end of the communication window.

5. **Message transfer**: Transfer messages as long as the communication window is available. The final transferred message may become fragmented. If this is the case, the DTN subsystem may either discard it, or transfer it partially.

In the following, we will outline each of these steps and discuss its relevance.

2.3.2.1 Successful neighbor discovery

As we will discuss in Chapter 4, nodes spend most of their time with their radio interfaces turned off. A neighbor discovery algorithm schedules short periods of activity during which nodes turn on their radio interfaces and send and listen for beacons.

This schedule, an assignment of active and inactive time slots within a defined maximum discovery latency, has to fulfill certain requirements: (1) The schedule for each node intersects non-trivially with the schedule of each other node. (2) Assuming the same energy availability, each schedule should have the same number of active slots. (3) Intersection between node schedules must be guaranteed, even when the clocks between the nodes are not synchronized.

As two nodes enter physical communication range, they will receive each other's beacons. When the first node receives a beacon from the second node, it stays awake and informs the other node of its presence. Once mutual presence is detected, both nodes are ready to exchange data. On a logical level, communication becomes possible.

2.3.2.2 Data freshness

As the two nodes are about to establish a communication channel, they check, if one of the node's information changed since their last encounter. This check, potentially partially integrated with the neighbor discovery itself, helps the nodes to distinguish, if it makes sense to invest energy and time in communication.

Let's assume the two nodes are carried by two travelers sitting in the same bus. Repeated data exchange with the same node, while no new payload data became available and routing information doesn't change, may needlessly deplete energy resources.

However, as new messages become available, forwarding them to the nearby node may be useful, especially when nodes may separate soon, as one traveler is about to exit a bus.

To ensure data freshness, nodes may keep a version counter which they update whenever files or other state changes. Including this counter in the discovery beacon allows nearby nodes to instantaneously decide if they need to a node again.

2.3.2.3 Exchange of routing information

Depending on the employed routing scheme, the nodes need to exchange various metrics used to predict future connectivity and possibly future energy and traffic availability.

Two trivial routing schemes are *flooding* and *direct delivery*. *Flooding*, also known an *epidemic routing*, copies each message to all encountered nodes. While this approach may not be feasible for many applications in practice due to memory constraints, it provides a theoretical lower bound in delivery speed, while not requiring any specific routing information.

Direct delivery only forwards a message, if it encounters its intended recipient directly. Again, this approach does not require any specific routing information. However, it offers a quality metric for routing schemes. If message delivery for a specific routing scheme takes longer than direct delivery in all but pathological cases, that routing scheme is infeasible for practical use.

Practical routing schemes consider to a varying degree the time-varying connectivity, storage and energy availability as well as queue length and traffic demand, using so-called *oracles*. The routing schemes make predictions about the future of these metrics. The better these predictions are, the more efficient the forwarding becomes. Oracles are highly specific to use cases, and differ greatly between human controlled *passive mobility* and active *self-controlled mobility*.

In this work, we present two routing schemes for human controlled mobility. One is based on the time-varying social network of a human, while the other also considers geographic location probabilities. Chapters 5 and 6 will discuss these in more detail.

2.3.2.4 Message carrier selection and prioritization

Fundamentally, message forwarding has two main flavors with respect to message replication: (1) *single-copy* and (2) *multi-copy* strategies. *Flooding* is inherently a multi-copy strategy, while *direct delivery* only keeps a single copy of a message in the network.

In the context of the handshake of the communication phase in a disruption tolerant network, once the nodes exchanged routing information, each node iterates over all messages in its message cache. Depending on the final recipient of a particular message, the routing scheme decides which of the current nodes is the better carrier for it. As contact opportunities may have limited duration, nodes may not be able to transfer all messages. Therefore, nodes also order messages for transfer according to their priority or utility.

Two messages addressed to the same recipient do not necessarily need to be forwarded along the same path. Load balancing may assign these messages to different nodes, especially, when their likelihood to reach the destination is similar. Also, as one message might have a lower priority or later delivery deadline, a slower but cheaper and more reliable path may make more sense.

2.3.2.5 Message transfer

As the handshake completes, the participating nodes initiate message transfer. While the nodes forward each message in their message store to its chosen carrier, the connection itself may break at any time due to mobility.

If a connection breaks, a message may become fragmented. In response, the nodes may either (1) discard the partial message, or (2) keep and potentially forward the partial message to other nodes. Both options have advantages and disadvantages.

Discarding partial messages is simpler to implement. As messages become larger, more bandwidth may be wasted if a connection breaks down. Therefore, discarding partial messages limits useful maximum message sizes, potentially leading to fragmentation in the application layer.

Alternatively, nodes may *keep partial messages*, thereby fragmenting a message. This leads to management overhead to keep track of which part of a message was already transferred. As a message is split into several parts, those parts may be forwarded along different paths. If one of the parts gets lost, the remaining parts may become unrecoverable, leading again to wasted bandwidth. To deal with this, the concept of custody transfer was developed. A node may forward messages, but only if a message was reliably reassembled at a particular node, this node may accept custody for this message, forwarding it further towards its intended destination. However, this process leads to higher implementation complexity and management overhead.

Once message transfer completes, nodes return to discovery mode and wait for state changes. As a node generates a new message, for instance through some ongoing measurements or user interaction, the handshake starts again. Alternatively, nodes move away from each other with each node carrying their respective messages towards their destinations.

2.3.3 Routing Metrics

Depending on the application scenario, the routing scheme will optimize different metrics, which may lead to different paths through a network, cf. Figure 2.3. Different optimization goals include (1) minimum delivery latency, (2) minimum cost, and (3) the minimum number of hops.

The most common used routing optimization goal in the literature is *minimum delivery latency*. A message is to be transferred to its intended destination as fast as possible. Success depends on how accurate future connectivity can be predicted.

However, if transfer of a message is associated with a cost, e.g. using micro-payments, an alternative optimization goal is *minimum cost*. This cost is not necessarily monetary, but can also be a management decision. The cost of transfer may depend on the available energy of a node, a node's popularity, but also how large a message

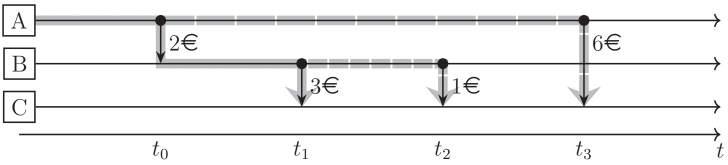


Figure 2.3 Depending on the routing metric, different paths through a network are possible. The *fastest* path passes a message from node A to node B at t_0 and from node B to node C at t_1. The *cheapest* path passes a message from node A to node B at t_0 and from node B to node C at t_2. The *shortest* path passes a message from node A to node C directly at t_3.

is. For specific messages, it may be cheaper to take a slower path, as long as they arrive within their deadlines.

As energy, storage, and bandwidth may be limited, a routing scheme may instead choose to *minimize the number of hops*. This metric is typically used in continuously connected, wired networks. The number of hops serves a stand-in for routing overhead to be minimized while at the same time providing an elegant safety net against routing misconfiguration in the form of maximum hop count after which a message can be discarded. However, in disruption tolerant networks, other metrics such as cost or delivery latency are typically more useful.

2.4 Connectivity Traces

To be able to evaluate different routing algorithms in a reproducible, reliable manner, we use simulation models. These simulation models can replay arbitrary node movement and emulate node behavior. This allows us to compare the advantages and disadvantages of a routing scheme for specific scenarios.

For the simulation model to have a realistic node mobility, there are two fundamental options: (1) an synthetic model that derives a node mobility from an analytical function, and (2) recordings of actual node mobility during a previous deployment.

A *synthetic model* has the advantage, that an arbitrary number of nodes can be simulated for an arbitrarily long time. However, the underlying assumptions might be overly simplified and therefore unrealistic.

In contrast, the recording of actual node mobility, a *connectivity trace*, especially from a real-world deployment, is realistic. This comes at the cost of a limited number of recorded nodes and limited runtime. Also, a trace may very cumbersome to record. However, more and more such traces exist. A useful repository for such traces is `http://www.crawdad.org`.

In the following, we will discuss a number traces, we used for our own simulations. Some of them only recorded connectivity, that is which node was in contact with which node at what point in time. Others also recorded locations of the nodes. In addition, we will discuss a trace and mobility model, we generated ourselves for the use case of wild life observation in its natural habitat.

2.4.1 MIT Reality Mining

In the context of the MIT Reality mining project [EP06], a group of 100 subjects at MIT where equipped with Nokia 6600 mobile phones over the course of the 2004/5 academic year for about 9 months. These mobile phones used standard Bluetooth scanning every 5 min to record all nearby active Bluetooth MAC addresses. The Bluetooth chipset is a class 2 device, which implies a range of about 25 m.

75 of the users were either students or faculty at the MIT Media Laboratory. A further 25 users MIT Sloan business school physically next to the MIT Media Laboratory. The MIDP2 Java application running on the phones recorded call logs, nearby Bluetooth devices, cell tower IDs, application usage, and phone status.

Overall, the data set on 100 subjects represents about 450 000 h of information on coarse user location, communication, and device usage. This data was used to recognize daily and weekly personal and organizational patterns. It is one of the most comprehensive records of social interactions available at present and is commonly used as a basis for evaluation for disruption tolerant routing algorithms, including [DH07].

2.4.2 Human interaction in the Haggle Project

The Haggle project [HCS+05] collected a similar data set [Jam06]. Over the course of the Infocom 2005 conference in May 2005 in Miami, FL, USA, 54 out of a total of about 800 conference participants were equipped with an Intel iMote, a type of a Bluetooth enabled sensor node in a rugged packaging. This sensors scanned for all nearby Bluetooth devices every 2 min. Of the 54 distributed sensors, only 41 resulted in useable data, with 11 nodes having various hardware issues with the batteries or the storage and 2 nodes gone missing.

Over the course of the conference, in addition to the 41 nodes, 182 external devices were recorded. Overall, the researchers recorded 22 459 contacts between the handed-out nodes, and 5791 contacts to external nodes. The maximum number of nodes a node recorded at the same time is 12, while the maximum number of external devices recoded at the same time is seven.

The researchers reported that the inter-contact distribution exhibits a strong heavy tail property, as an approximate power law distribution for the time frame 2 min to 1 d, with a coefficient of 0.4. The reported patterns are very similar to previous data sets for corporate and academics environments [CHC+05]. Hui et al. further report that a power law mobility profile for inter-contact time with a coefficient of lower than one, while common in the real world, is problematic for existing routing algorithms.

The evaluation of this mobility trace became the source of the BubbleRap approach by Hui et al. [HCY08]. Aside from being a useful source for node mobility information, this mobility trace reiterates the importance of energy efficient but fast neighbor discovery, as the vast majority of the time is spent in the discovery phase.

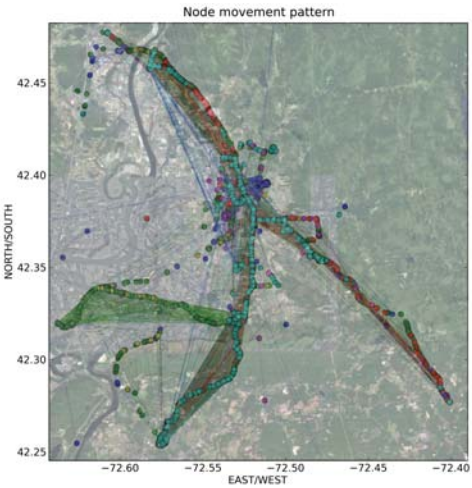


Figure 2.4 Graphical Representation of the area covered by the UMASS DieselNet. This includes the GPS locations of 33 buses recorded every two minutes or on contact events with another node around Amherst, MA, USA.

2.4.3 Dartmouth Outdoor Experiment

One of the most used synthetic mobility models for MANETs is a random waypoint simulation. Basically, randomly choose a random destination in the simulated area and then move to that destination at a random speed. Once a node arrives at this waypoint, the process repeats.

In the Dartmouth Outdoor experiment [GKN+04], Gray et al. recreated the scenario on an athletic field of 225 m by 365 m with actual people. Participants were given 40 WLAN and GPS enabled laptops and instructed to walk in the field randomly. One additional laptop served as a control station. The location was recorded every 1 s synchronized via the GPS clock. Every 3 s, each laptop broadcasted its location via WLAN.

In addition, the experimenters used a traffic generator, sending about 423 bytes of data traffic including header per laptop per second. This traffic was chosen, as it corresponded with the traffic in the author's trial runs of prototype military applications and thus deemed realistic.

As all participants met in a central location at the beginning and at the end, only the second and third quarter of the trace resembles the random waypoint model. The trace, available at [GKN+], allows us to compare the behavior of routing algorithms in a semi-realistic human mobility on an open field.

2.4.4 UMass DieselNet

While the previous mobility traces are based on human based mobility, this trace [BLM+08] is based on buses around Amherst, MA, USA. 30 to 40 buses were equipped

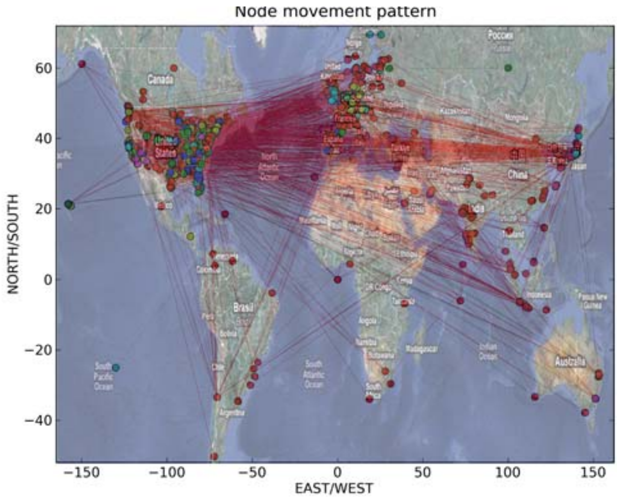


Figure 2.5 Movement of 221 very active BrightKite users over two month. We recorded this trace as an example of global human mobility. For the purpose of our later evaluation, we are assuming that all of these users participate in a pocket switched network.

with a custom transceiver package consisting of a HaCom Open Brick computer (P6-compatible 577Mhz CPU, 256MB RAM, 40 GB hard drive) running Linux with one IEEE 802.11b interface scanning for other nearby buses, and an additional IEEE 802.11b interface providing an access point to the bus passengers. In addition, each bus recorded its GPS position.

Aside from the town of Amherst, the area serviced by the busses includes some sparsely populated nearby areas and has an overall size of approximately 400 km^2, see Figure 2.4. For our later evaluation, we used the Mass DieselNet traces from the fall of 2007. They contain GPS movements of 33 nodes for 18 days, where we assume a simulated radio range of 150 meters.

2.4.5 Self-reported global mobility in Brightkite

BrightKite is a social networking website where user can share their current location. It provides an API to download user data including all locations. We observed the 221 very active users over a period of 2 month and converted the location traces into connectivity traces under an assumed radio range of 300 meter. The resulting network is shown in Figure 2.5. Overall, the process resulted in 43786 network events.

We use this trace as an example of global human mobility. Traces for regional or global mobility are difficult to obtain for larger groups. Therefore, for our evaluations, we relied on self-reported, publicly accessible mobility traces of a social networking site. To preserve the privacy of the users, we removed all personal identifiers and are only reporting aggregate data on the results. The location information, which is necessary to evaluate location based routing schemes as discussed in Chapter 6 and which may still allow the re-identification of users, remains confidential and only accessible to the author and his collaborators.

2.4.6 N4C trials mobility data

N4C, short for *Network for Communications Challenged Communities*, was an EU project within the *Future Internet Research and Experimentation (FIRE)* initiative with the EU 7th framework theme. Under the leadership of the Trinity College in Dublin, Ireland, 12 partners developed new paradigms and experimental facilities. The project, running from May 2008 to April 2011, included a six different experimental trials in remote areas of the Padjelanta national park in Northern Sweden and in Kočevje in Southern Slovenia in summer and winter.

While the overall project provides good insight in the requirements, design, and applications of disruption tolerant networks, we focus here on the trials in the summer of 2010 in Northern Sweden [FMM+11]. Farrell et al. successfully used existing helicopter routes to carry email and web data to four remote *village DTN routers*. These solar-powered Wi-Fi routers to be used by nomadic Sámi reindeer herders are between 20 km to 50 km from the nearest communication and power infrastructure and intermediately store requests that are forwarded via the helicopters using the *bundle protocol*, one of the most commonly used DTN protocols.

The researchers not only make the hardware and software available with detailed specifications and source code, but make all non-private data available for download and analysis. This includes all connectivity traces, all generated message sizes and times, message delivery rate and delay, as well as trace routes. As an additional service to the engineering community, they also included detailed failure and problem reports [FMM+11, p. 26]. Reasons for data loss included a design flaw of the used system-on-chip, using a RAM disk as the root file system, resulting in logs being lost on power loss, and other `logrotate` misconfigurations. Device specific errors included full storage devices from logging of a very fluctuating GSM uplink and bugs in `dtnd`, a routing daemon for the bundle protocol.

The overall deployment, including startup, operation and tear-down, lasted 60 days. Uptime of the routers ranged from 29 days to 43 days. There were 85 unique web requests, of which 49 % were successfully delivered. 9575 contacts were recorded during the trial run. Aside from the contact information itself, this data are a strong indication for the requirement for good neighbor discovery algorithms as evidenced by the long time between useable contacts in real-world deployments.

2.4.7 Underground radio propagation in RatPack

The *RatPack* project, running at COMSYS and Tübingen University from 2007 to 2010, aimed at observing the behavior of the common Norway rat, *Rattus Norvegicus*, in its natural habitat. While the common Norway rat has long been used as a model organism in the field of life sciences, comparatively little is known about their natural behavior as they dwell in underground burrow systems. Previous research focused mainly on archeology-like approaches, digging out existing burrows and determining

Figure 2.6 Norway rat equipped with sensor node. Our first prototype was based on the *mica2dot* sensor node, sending data using a low-power CC1K based radio in the 433 MHz part of the ISM band. Later prototypes operated based on the CC2420 radio in the 2.4 GHz band.

behavioral patterns based on found remnants. In this project, we equipped rats with sensor nodes, which form a loosely connected, often interrupted network—a disruption tolerant network was needed.

Our first setup consists of a mica2dot node powered by a coin cell battery and supporting a number of custom made sensor boards, see Figure 2.6. In our laboratory prototype each node recorded data (e.g. data from accelerometer and ultra sound) until its memory is exhausted and subsequently transmits it to the base station when signal quality is sufficiently high. While this approach was feasible for preliminary test scenarios in a controlled laboratory environment, this setup is infeasible for a deployment in a real-world burrow system, because of the short battery lifetime, restricted radio range, and size.

To validate the range of typical sensor networks, we performed a range of experiments. While the outdoor range of CC1K and CC2420 radios was about 70 m to 90 m, both only achieved a range of about 1 m in an artificial linear rat burrow, we built on a field near the university in Aachen. The artificial rat burrow had a diameter of 0.1 m, that we kept open using drainage piping, to mimic the actual size of a typical rat burrow [Cal63]. Sensor nodes were put into a plastic bag and connected to a long rope, at which they were pulled through the artificial burrow.

In order to evaluate different routing schemes for the RatPack project, we built a rat burrow simulator. This simulator provided us with location and contact traces based on our initial measurements.

2.5 Acknowledgments

The idea for the *RatPack* project developed in 2005 and 2006 during the Master's thesis of the author with significant input from Johannes Thiele, Hanspeter Mallot, Olaf Landsiedel and Klaus Wehrle at the University of Tübingen. Initial DTN prototypes were developed with the help of Nicolai Viol, who first started as a student assistant in 2007 and later worked on his Diploma Thesis on DTNs [Vio09] in 2008 and 2009 at RWTH Aachen University. The sensor node prototypes were developed together with Okuary Osechas and Johannes Thiele in the context of the RatPack project funded by Microsoft Research. The rat burrow simulator was developed in the context of the Diploma thesis of Gregor Fabritius [Fab09] on path reconstruction and localization in underground burrow systems. The BrightKite mobility trace tool was developed in the context of the Diploma thesis of Daniel Schmitz [Sch10a] on geographic routing.

3

Predicting Experience in Disruption Tolerant Networks

Disruption Tolerant Networks (DTNs) are designed for environments, where traditional networking paradigms do not necessarily apply [Fal03; RBK07]. However, even in conventional circumstances, service developers want to guarantee a minimum level of service quality to the end-user, which may not always be achievable using conventional methods due to constraints in terms of network nodes, paths, and connectivity. So, while the community is on a quest for the DTN killer app [LH09], we argue that most networked applications, especially in a mobile environment, would highly benefit from becoming aware of delays and disruptions, and varying connectivity over time.

Failure to fulfill basic user expectations in terms of application quality will lead to the service not being accepted. Therefore, satisfying user expectations of performance and reliability is a key factor for the success of services and applications, especially in challenging network environments.

To ensure end-user satisfaction, we need to first identify the limitations of such scenarios. This is not restricted to network characteristics, but must also include service requirements as well as differences between different types of users. Since a users' subjective quality perception is an integral part of the success of a service, analyzing network services with traditional Quality of Service (QoS) metrics is not sufficient. Instead we need to introduce the concept of *Quality of Experience (QoE)*.

QoE is currently defined by the International Telecommunication Union (ITU) as "overall acceptability of an application or service, as perceived subjectively by the end-user"[IT07].

Using this definition as a basis also for communication in challenging environments, we need to define which features in challenged networks affect the acceptability of an application or service by the user. Parameters such as unpredictable link availability and long delays, are well known properties of challenged networks. Each of these parameters has a different influence on the user's QoE.

3.1 Our Contributions to QoE Prediction in DTNs

In this work, we argue that network structure and QoS parameters are insufficient to evaluate services in challenging environments. The requirements implied by the service's communication structure, but also the specific type of user trying to access these services are significant factors. All these influence the actual outcome for the user, her Quality of Experience.

Our contributions are:

1. **Identification of characteristics:** Services have different levels of chattiness and application expectations. Networks can be differentiated based on link reliability, fluctuant transmission speed and other characteristics. A user's expectations depend on her knowledge of the underlying systems.

2. **New metrics for QoE in DTNs:** To evaluate the Quality of Experience for a service, we need to consider metrics in the network, based on the newly identified characteristics.

3. **Guidelines for assessing QoE in DTNs** Using the developed metrics and characteristics, we offer guidelines, when an application might be particularly well-suited for challenging environments. This in turn gives hints for managing user expectations, when network conditions are not met.

3.2 Service Differentiation

Not all network services are equally suited for challenged environments. There are a range criteria that may help in deciding whether a service is suitable for a particular scenario. In the following, we analyze a sample of example services and derive criteria based on their specific communication structure.

3.2.1 Browsing and Web Surfing on Wikipedia.org

As our first example, we consider the digital encyclopedia *Wikipedia*, as one of the most popular website, consistently around place 7 of the most popular websites [Ale16], only surpassed by popular search engines, Facebook, and Amazon. It is the single most popular reference and educational resource on the web. There are currently 292 different languages supported [Wik16], with 133 having more than 10 000 articles making it also one of the most internationalized resources.

As a conventional web application, it uses the HTTP protocol with vanilla HTML. It exhibits a couple of properties that make it well suited for communication via challenged networks, in particular: (1) An article in Wikipedia and its resources can easily be bundled, and (2) communication is not continuous but bursty.

3.2.1.1 Data Prefetching and Bundling

Wikipedia articles consist of a main text document, the HTML source, and a variety of included resources such as images, style sheets, scripts, or embedded audio or video files. The included resources form an integral part of the article, making it a safe assumption that the user is interested in all the supporting resources. Therefore, all files can be combined into a single data bundle.

With current standardization efforts acknowledging the need to reduce the effect of high latency connections, even standard protocols like HTTP and HTML become more aware of network speed.

Current standardization in `HTTP/2` [RFC7540] aims at solving the same problem, using *push promises*. As the server can anticipate, which resources the client will likely need—it serves the underlying HTML after all— it can send them to the client automatically. This reduces the number of roundtrips significantly, leading to a better quality of experience of users on a high latency network.

Similarly, `HTML` standardization aims at including *preloading* [Ily16a] and *resource hints* [Ily16b] that allow websites to specify very early in the connection which resources a critical for a website or which other websites a user is likely to navigate to next, using either HTTP `Link:` headers or `<link>` elements in the HTML header In turn, the user agent may already prefetch these resources or likely next websites, making them available virtually instantaneous when the user clicks a link.

However, both these approaches depend on an interactive connection between the client and the server. Data bundling would allow to push the whole article in a single bundle and reduces the need for additional files or server roundtrips altogether. A generalization of this approach on the network layer led to the development of the *Bundle* protocol [RFC5050], a commonly used DTN protocol. This protocol allows pushing bundled application payload without requiring a synchronous end-to-end.

3.2.1.2 Bursty Communication Style

Interacting with the Wikipedia site typically involves requesting an article and reading it. After a while, a user may request other articles linked to from the original text. Often related articles will be loaded in a background tab.

This corresponds well to general web browsing behavior. According to Kumar and Tomkins [KT10], the median time between successive page views is about 12 s, measured through data recorded using a browser toolbar. They suggest that multi-tab and multi-window browsing, as well as often using the back button contributes to this short time, but did not log any information to support this hypothesis. However, the median time between referrer and referred to pages was 18 s and the 90th percentile was 5.2 min. The median of total page views per week was 113, with 90th percentile being 1097 page views per week.

This data supports that in web surfing, communication happens in short bursts, as opposed to live audio or video conferencing, which require a continuous, stable connection. In a scenario with limited connectivity, a request for a referenced article can be handled in the background while the user is still reading the referring page using techniques like the above mentioned preloading. Once the user finished reading, chances are that the follow-up page is already completely loaded in the meantime, even if the network experienced variable bandwidth or delay.

3.2.2 Collaborative Event Scheduling

Our next example is a class of collaborative event scheduling websites, such as Doodle, a popular web service used for coordinating meetings among multiple participants. Doodle presents every participant a poll with multiple possible options for a meeting date and location, asking the user to rate every option as good, acceptable but not preferred, and bad.

Similar to Wikipedia, Doodle is a conventional web application, which uses the HTTP protocol with vanilla HTML. It exhibits a different set of properties, that also make it well suited for communication via challenged networks, in particular: (1) Timing constraints for event scheduling requests and answers are soft, and (2) the order of the delivery of answers is not particularly important.

3.2.2.1 Soft Time Constraints

While the implementation of Doodle, due to its dependence on HTTP and therefore TCP and IP has relatively strict constraints with respect to the underlying network connectivity, from a user's point of view, timing constraints are much more relaxed. The service itself does not impose constraints on a time to participation, only the poll initiator does by setting some limit on what she considers a timely response.

Therefore, if a poll initiator is setting a deadline of 2 days for the response to a poll, it does not matter, if a participant receives an invitation to this poll within milliseconds or within a few hours. If the invitation is a self-contained data bundle, which contains all necessary resources for displaying the poll and dispatching an answer bundle, it would be functionally and visually equivalent to the current implementation for a user.

3.2.2.2 Lenient Delivery Order Constraints

Replies to a poll—for the time until the poll closes—are independent of each other. The initiator can tally all replies, potentially weighing the replies of specific respondents stronger, at the end of polling time. It does not matter for the correctness of a poll result which reply arrives first.

We call this property *lenient delivery order constraints*. Poll replies can logically only arrive, after a poll request was sent out. However, there is only one hard deadline—the end of the polling time. Any reordering before this deadline is inconsequential.

Packet order requirements in typical transport layer protocols are handled in an all-or-nothing way. Either a protocol is entirely stateless, where no restrictions on the order are imposed at all, or a protocol tries to ensure absolute packet order in a stream with respect to protocol layers above it. The canonical examples for this

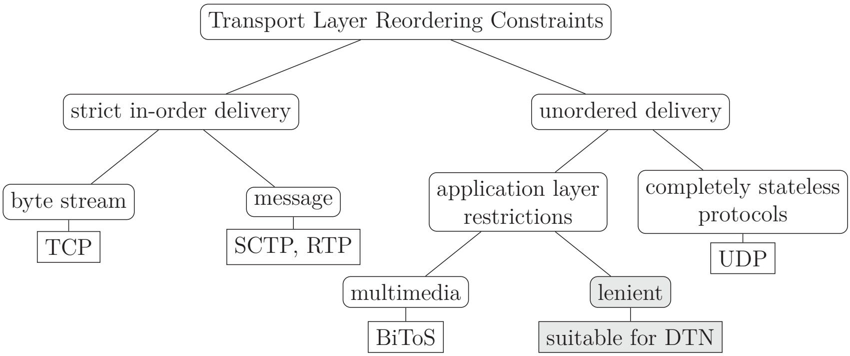


Figure 3.1 Different classes of ordering constraints. While TCP ensures a byte stream with in-order delivery, UDP makes no guarantees. From an application perspective, the more lenient it is, the more suitable it is for DTNs.

are UDP for a stateless and TCP for a stateful, order ensured protocol. Metrics used for measuring packet reordering in a network, e.g. as defined by the IETF [RFC4737], are then used to tweak implementation and deployment parameters for ordered delivery protocols.

However, in terms of application requirements, delivery order requirements that do not fit into these two categories are not yet well formalized, the exception being multimedia streaming applications, such as BitTorrent Streaming [VIF06]. While a media player is buffering media for a running presentation, chunks that are closer to being presented have a higher priority. However, other chunks are downloaded based on their rareness in the network and therefore out of order.

Nevertheless, as we see in the collaborative event scheduling example, this type of lenient delivery order constraints is useful also for non-multimedia applications, see also Figure 3.1. The more lenient an application can be, the better it is suited for challenged networking environments.

3.2.3 Web Syndication of Blogs and Media Files

Our third example class of applications is blogs, as the short form for weblogs. On blogs, individuals or groups regularly post articles or media files, such as news and current events, but also personal thoughts and opinion pieces. These articles are then published on a web server.

As for the readers of these blogs, they regularly visit these websites to check if there is new content. However, especially for smaller blogs, updates only happen irregularly. So, most of the time a reader would not find new content.

By publishing a so-called feed, e.g. using `RSS` or `Atom`, the blog owner can free the reader from manually checking for new content. A user would use a feed reader or feed aggregator, in the case of audio a *podcatcher*, that regularly checks a feed for updates; she *subscribes* to a feed. The feed reader can then automatically download the new content and prepare it for consumption by the user.

Before the advent of web syndication, mailing lists and newsgroups served a similar purpose. While they still exist, many users do not want to disclose their email address to avoid spam. Also, the automatic integration into feed readers and podcatchers makes their use very convenient.

3.2.3.1 One-Way Communication

Once a user subscribed to a feed, the delivery of an article or a media file is semantically a one-way communication. While technically, a feed aggregator regularly pulls a web server for updates, this is specific to the used web syndication mechanism. In particular, the user does not need to interactively exchange data with the web server to make use of the service.

Non-interactive services are arguably the most well suited type of service for distribution via disruption prone, challenged networks. In the context of challenged networks, the publishers of such services generate data and push it into the networks gateway nodes. From there it can be distributed to other clients without the need of a communication channel back to the data source.

Likewise, the pushed data does not depend on some characteristics of the receiving client. All clients that subscribed to a particular feed are semantically equal. This can simplify routing decisions significantly, allowing a fallback to broadcasting or multicasting.

3.2.3.2 Hop-to-Hop Usability

As all clients that subscribed to a particular feed are semantically equal, the pushed data is potentially directly useful for every user on the forwarding path towards a subscriber, if she is interested in the content. With the content being self-contained, without requiring additional external resources, she can just load a transferred media file into her own player.

This usability of forwarded content at every node on the forwarding path can become a powerful incentive for users to participate in a disruption tolerant network. A user does not provide resources purely altruistically, but has a direct selfish advantage from participation.

Now, while from a technical perspective, the self-containedness of the distributed article or media file is enough to make it hop-by-hop usable, there need to be appropriate legal conditions for such a usage. The main conditions are: (1) usage rights, in particular the permission to redistribute the work and (2) privacy protections.

Many publishers of popular podcasts explicitly license their work under a Creative Commons license which explicitly allows redistribution. For example, escape pod licenses all content under a Creative Commons Attribution, Non Commercial, No Derivatives license [Esc16], which can be summarized as follows: As long as a user does not modify or sell the content, she is free to redistribute it with appropriate attribution.

So, while one can argue that simple forwarding of content to its originally intended recipient, would not allow an intermediary to make the content available to someone else for consumption aside from possibly some legally guaranteed exemptions that might apply, such as fair use. Explicitly attaching a license that allows redistribution makes the content creators intention clear, that she agrees to such a practice.

As to privacy implications: If a node on the communication path cannot only inspect forwarded messages, but even makes the content potentially available to the owner of the node, there need to be protections in place to ensure the privacy of the original subscriber of a feed. However, these are not the focus of this work.

All in all, if forwarded content is usable at every hop of a path, the corresponding application provides an incentive mechanism that makes it more suitable for usage in disruption tolerant networks.

3.2.4 DTN Antipatterns for a Service

While we identified a range of characteristics that make services well suited for usage in challenging network conditions, there exist characteristics that preclude specific classes of services from being useful in a disruption and delay prone environment. These characteristics include: (1) Chattiness and (2) delivery rate requirements.

The other way around, if a service scores low on chattiness or can deal with lost packets, it may be well suited for disruption tolerant networks. These metrics are gradual, so there is no fixed threshold. Rather, they can be regarded as a hint, which services are more likely to be successful and which are not likely to be successful.

3.2.4.1 High Interactivity Pace/Chattiness

This metric captures how nodes running a specific application or service interact. Interaction patterns can be separated into different classes: (1) Session-less, (2) session-oriented, or (3) continuous [OK04b].

Session-less, one-shot operations are well-suited for challenging environments. Single packets might be *nullipotent*, meaning that they do not change the state of the recipient, or *idempotent*, meaning that it does not matter if a packet is received multiple times or only a single time. Non-idempotent one-shot operations may be problematic, in a challenged networking environment.

Session oriented protocols require multiple round trips between a client and a server. An example of a session oriented transport protocol is TCP. The initiating client first sends a `SYN` packet to a server. The server responds with a `SYN/ACK` packet back. The client confirms the opening of the connection with an `ACK` packet. Up to this point, the client and the server did not exchange any payload data. If the connection needs to be secure, the client and the server would perform a TLS handshake. This TLS handshake requires at least two additional roundtrips with still no payload data exchanged between the client and the server. Performing this type of nested handshakes over a disruption tolerant network may become prohibitively expensive.

Finally, services such as video conferencing or IP telephony require both a timely in-order delivery of their packets and a high constantly available bandwidth. ITU

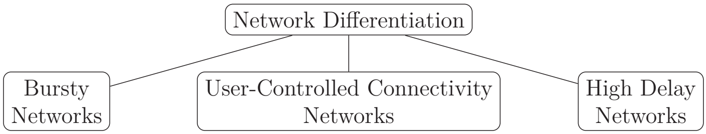


Figure 3.2 Different classes of networks based on predictability. Networks range from fast changing networks with little to no predictability, to user controlled probabilistically predictability, to exactly predictable connectivity.

recommends a one way latency of below 150 ms [IT03]. As such, it is physically impossible to run these types of services over delay and disruption tolerant networks.

These requirements on the pace of the communication are inherent to the nature of such services. As such, services that require timely roundtrips between communication partners are not well-suited for DTNs and will not be able to deliver a satisfactory user experience.

3.2.4.2 High Demands on Delivery Rate

Related to the previous point of protocol chattiness, if a service has high demands on the delivery rate of messages, it may be unsuitable for challenging network environments. The acceptable packet loss for services varies from 100 % for remote shell-like applications down to 20 % and lower for statistical sampling applications. In Voice over IP, which is unsuitable for DTNs for latency and chattiness reasons, a packet loss of about 1 % to 5 % is tolerable to achieve acceptable voice quality [MTK02], depending on the codec used.

The question of demands on the delivery rate boils down to how the API of a service is organized. If requests are *nullipotent*, the service may not suffer from packet loss. *Idempotent* requests may be more difficult to handle as at least a single delivery is required. Packet duplication becomes a significant issue, if a service request is neither nullipotent nor idempotent.

In consequence, it becomes also important how a byte stream is packetized. If each individual packet includes only self-contained requests, they may still be useful, even if an API request in-between may be lost. However, if a request becomes fragmented into several transport or network layer packets, that may be dropped independently, this becomes more problematic.

3.3 Network Differentiation

Disruption and Delay Tolerant Networks are a broad category used to describe a variety of networks in challenging conditions. Examples of such networks include sensor, cellular and military networks [Fal03], see also our more detailed discussion in the previous chapter in Section 2.3.1.

While the range of possible use cases and network conditions is wide, some of the common properties of such networks are:

Figure 3.3 An example for a bursty network. A train driving through radio cells may have reliable and fast connection while in range of an access point. However, the connection may go down at any time without prior notice.

- **Unreliable links:** Even when a direct link between two nodes exist, it may drop at any time, e.g., because of node mobility. In addition, a link may be asymmetric.
- **No end to end path:** DTNs typically do not rely on stable end to end paths between all nodes of the network. Growing network complexity and a rising number of participating nodes exacerbates this problem.
- **Sparse connectivity:** Depending on the topology and mobility of network nodes, nodes might not be able to communicate over extended periods of time.

In our analysis, we group these networks into three classes with certain common properties as well as substantial differences. Later we will be able to map these classes to specific application needs to identify combinations potentially well suited for a DTN scenario from a quality of experience point of view.

- **Bursty Networks:** These networks have fast changing connectivity. Network links may be robust and reliable while available but may go down at any time without prior notice.
- **User-Controlled Connectivity Networks:** In this category, the communication nodes are piggy-backed to a user, whose behavior is only predictable to a limited extent.
- **High Delay Networks:** In these networks, a link may have a long delay or a limited capacity, but both are very predictable, e.g. in inter-planetary communication.

In the following, we will describe these network categories in more detail.

3.3.1 Bursty Networks

A bursty network is characterized by a rapidly changing connectivity pattern. Such networks are often found in scenarios where the user moves through an area with partial connectivity. Examples of this include cars driving through radio range of multiple WLAN access points (e.g. [OK04a]) or similar Mobile Ad-Hoc Networks (MANETs).

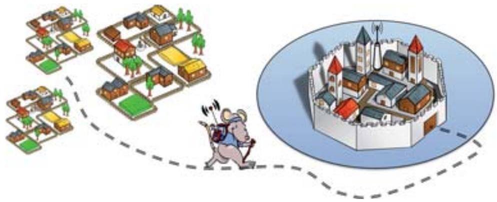


Figure 3.4 An example for a user-controlled connectivity network. A villager on his way to a city ferries messages from a rural area to and from an area with global connectivity. However, the mobility and therefore the connectivity and timing is user-controlled. The network node carried by the user may probabilistically predict future connectivity but has neither control nor guarantees.

Network links are robust and reliable while they persist but may go down at any time without prior notice. When a new link is established, it may be in a completely new environment, e.g. with a different carrier or in a new IP address subnet. Figure 3.3 shows such a scenario, a train traveling through various radio cells.

Shirky [Cla03] makes the argument, that this currently is the state of mobile networks and will most likely remain so for the foreseeable future. He argues that the economics of connectivity dictate this development. While people claim they want a ubiquitous and convenient network, what he calls a *perma-net*, they are not willing to pay for this, so it becomes economically infeasible for providers. According to him, what people actually need and use is a *nearly-net*, an “archipelago of connectivity in an ocean of disconnection” [Cla03].

His guiding example is the phone installed in airplanes seats. In principle, the system is permanently available, but calls are prohibitively expensive. Mobile phones on the other side are only connected intermittently, but significantly cheaper. Operators count on the users being willing to pay a significant premium for ubiquitous connectivity. However, as users are able to defer calls to later when the cheaper option becomes available, most will. In turn, phones in airplane seats are economically not viable. In consequence, especially outside of population centers, the resulting connectivity will remain a *nearly-net*, that is a burstily connected network.

3.3.2 User-Controlled Connectivity Networks

With communication nodes piggy-backed to a user in this class of applications future connectivity can only be predicted to a limited extent. Therefore, this category is related to the previous class of bursty networks.

However, mobility is controlled by the user. Through this, the user indirectly controls the availability of network connectivity. Connectivity may be predictable using probabilistic models, but there are no guarantees. In the research community, most mobility traces are based on this type of networks. Therefore, these traces are the most commonly used basis to evaluate DTN routing algorithms.

Figure 3.4 depicts such a scenario, a villager living in a village without connectivity, but visiting a bigger city every once in a while [JFP04]. In such networks, nodes can

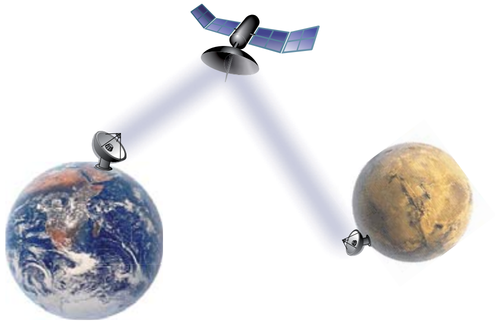


Figure 3.5 An example for a high delay, interplanetary network. A rover on Mars collects data and measurements. Once an orbiter—whose orbit is known in advance—is in range, the rover forwards the data to the orbiter. As Earth becomes visible to the orbiter, it forwards the data to receiver stations on Earth. Due to the speed of light, a message will take between 3 and 21 minutes to reach Earth. However, this delay is very predictable.

predict connectivity, if necessary with the help of the user. Services or applications running on the nodes can then use these predictions to alter their communication pattern.

The RatPack project discussed in Chapter 2 in Section 2.4.7 is a similar scenario. However, in this case, the connectivity is controlled by the animals under observation. This makes a prediction of future behavior more difficult.

Other scenarios include sailing vessels. While marine radio connectivity using VHF has a range of up to 40 km and marine shortwave radio can have a range beyond 5000 km depending on the used frequency and transmission power, high bandwidth connectivity to a coastal station is only possible in close range to the coast line. Location and therefore connectivity is controlled by the ship's captain and does not necessary follow paths that are optimal for communication.

3.3.3 High Delay Networks

These types of networks are the prototypical example of delay—as opposed to disruption—tolerant networks. The delay typically stems from the law of physics, as is the case in interplanetary networks or from other well-defined and reliable conditions, such as train schedules.

The first example is interplanetary communication [BHT+03], see Figure 3.5. The Curiosity rover on Mars collects data following a well defined mission plan. As soon as a Mars orbiter, such as the Mars Reconnaissance Orbiter, comes in in range, the rover can forward its data to the orbiter at up to 2 Mbit/s. Mars orbiters have a line of sight to Earth around two thirds of the time and can send data at a rate of 0.5 Mbit/s to 4 Mbit/s.

However, limited by light speed, a message will take between 3 min 7 s and 20 min 57 s to reach a receiving station of the Deep Space Network (DSN). This delay precludes the use of standard protocols, such as TCP [WPPSM07], but is easy to predict. As

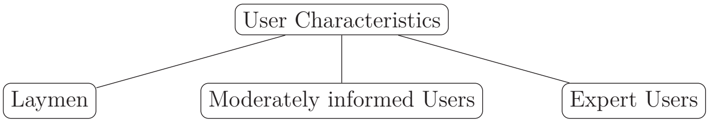


Figure 3.6 Different classes of users based on their knowledge about the underlying network. In general, the more informed a user is about the underlying mechanisms, the less frustrated she will be about artifacts arising from the underlying network.

the orbits and distances but also traffic patterns and available storage and energy resources of intermediary nodes are well known, the optimal route can be calculated.

An alternative, more terrestrial example is the use of trains and busses running on a well defined, reliable schedule [BGJL06a]. The fundamental difference to the previous group is that connectivity is influenced by an outside force out of the user's control, in the examples light speed, occlusion from orbital motion, or a train schedule. However, the connectivity, availability, bandwidth, and traffic is predictable with very high certainty.

3.4 User Characteristics

Since perceived service performance is very subjective, QoE cannot be defined without taking a user's specific situations into account. After we defined service and network characteristics in the previous sections, we now consider the influence of a user.

Several factors come into play, including previous experiences as well as situation-related factors. Knowing and considering the average users expectations and experiences with the specific technology in question will lead to a more precise QoE projection. Factors such as the user being a person that is open to technology, using modern technology on a daily basis or familiarity with the software and hardware used must be respected. These factors all may change over time, but not as rapidly and as extreme as other factors that we will consider in Section 3.5.

- **Laymen:** These types of users lack the understanding of the involved technologies. When they have used similar services via a reliable connection, they might become very frustrated when exposed to challenging networking environments.
- **Moderately Informed Users:** These users have a basic understanding of the underlying technology. While not aware of all the details, they understand the limitations and challenges of a challenging networking environment.
- **Experts:** They are well educated with respect to the underlying network technology and understand its limitations very well.

3.4.1 Laymen

The main characteristic of this first group is a fundamental lack of understanding of the technologies involved. Typically, this group is not very forgiving with service failure or delay and might become very frustrated if things do not work as expected.

The reason for this lies that they may have used functionally similar services via a reliable connection before. In the challenging networking environment, they are experience this supposedly same service but with severely degraded performance, without understanding why this happens.

As the perceived lower quality comes from the fundamentally different characteristics of the underlying network, it is important to design an appropriate user interface that exposes the underlying state and educates the user and manages her expectations. This class of users includes a broad range of groups, among them senior citizens and younger children.

3.4.2 Moderately Informed Users

In contrast to the former group, these users have at least a cursory understanding of the underlying technology and have a general understanding of the related limitations. They might be using related technology in their daily live and have adjusted their expectations based on the environment they use it in. While such users may still give up rather fast if something *just not works*, they are in general more pardoning regarding technical differences.

These users are aware of the basic principles of the underlying core technologies, such as the radio cell structure in mobile phone communication. However, they may not be familiar with the exact local details that influence the performance of a service. Understanding the presence and limitations of a challenged network in a given situation they might not expect the same quality of service with which they are familiar for that service via a reliable connection.

3.4.3 Expert Users

Finally, this group consists of the most educated users with respect to the technologies involved in a DTN. This includes users such as the developers of a particular service as well as people with a pronounced interest in the underlying networks.

With this understanding, they are very aware of the limitations, limited availability and performance conditions of networking in challenging environments. As this performance is in many cases dependent on the behavior of the user, she may even change her behavior to improve the service.

Networks depending on user controlled mobility are a good example of this. The expert user may adjust her routes to come into contact more often with either the challenged, partitioned part of a network, or a base station with global connectivity. With these improved contact opportunities and message ferrying capacities, the network performance improves.

Currently, most if not all DTN deployments are used exclusively by expert users. While this is expected for experimental technology, if DTNs are ever to become main stream, researchers need to look at user interfaces and interaction patterns, that allow also other users to use this technology productively.

3.5 Situational Influences

As previously mentioned, we need to consider not only the user's general attitude towards technology, but also take into account the specific situation. The mood and pressure under which a user operates a device or service heavily influences her perceived service quality. Therefore, it has a big influence on the user's QoE.

Assumptions over the mood and other situation-imposed influences are hard to predict in most cases and therefore can only be taken into account to a certain extent. Still, the nature of certain services such as disaster recovery services imply a certain user situation by their inherent nature [HCS+11]. While disasters lead to elevated stress and anxiety, people will also be very aware of the degraded performance of public infrastructure such as network connectivity.

We propose that these influences should be considered when evaluating a service and drawing conclusions from a field test. A user in a relaxed mood will act utterly different then the same user in a stressful situation.

3.6 QoE Metrics

We now define a set of metrics that help us to analyze network protocols and services that operate in a challenged networking environment. Together with the classification of user and network influences in the previous sections, these metrics allow us to predict the user's perceived quality of experience.

- **Speed of Interaction:** The speed at which users react to notifications can be an indicator of their satisfaction with a service. This depends on the application type and user context.
- **Comfort and Frustration Level:** While network protocols and applications do not necessarily raise the mood of a user, they should not be cause for additional frustration. Accordingly, this metric tries to capture the change in comfort or frustration.
- **Delivery Effectiveness:** This measure captures how many received packets actually had an influence on the user experience, as some packets might become obsolete if they arrive out of order or delayed.
- **Intercontact Times / Idle Period:** This metric captures the topology and workload of a network.

During a deployment, these metrics can then be measured to compare the QoE predictions with the actual user data. This allows a developer to improve the specific service in the average use case situation.

3.6.1 Speed of Interaction

The user's feedback is the most significant indicator of success for a given service. A user enjoying the service will most likely interact often and answer fast to upcoming requests or notifications. On the other hand, a user not reacting swiftly must not necessary be an indicator for a bad user experience. The nature of the service usually allows to define a certain range in which a user response can be expected. Measuring the speed of interaction should be done application-specific, for example by tracking the time between an incoming event that is signaled to the user and the time the user reacts to that signal.

3.6.2 Comfort and Frustration Level

A very subjective, yet meaningful indicator is the user's change of mood or comfort level during usage of the service. A good service performance should at least lead to a constant mood. However, services and performance experienced as bad will most likely lead to ignoring the application or even frustration by the user. Measuring the comfort level is a difficult task since it can hardly be done by technical means. Instead, developers will need to revert to traditional QoE measurement techniques such as questionnaires.

3.6.3 Delivery Effectiveness

For reasons lying in the nature of DTNs, not all packets dispatched into a challenged network will reach their destination. In turn, not all packets may be required for a satisfying user experience. Just as well, not all delivered packets may prove useful. There are scenarios where delivered packets cannot be used anymore because they are delayed or delivered out of order.

In DTNs the QoS metric *delivery ratio* must be extended by a *delivery effectiveness* stating how many delivered packets are actually used for service delivery and not dropped for the aforementioned reasons. Again, this metric cannot be measured at the network layer alone. The application itself must be aware of being in charge to provide feedback how many received packets were actually useful.

3.6.4 Intercontact Times / Idle Period

Depending on the network structure, packets may not be transported from one hop to the next for a significant amount of time. If a connection between two neighbors is established it is still possible that a packet is not transported for a number of reasons. Such reasons include limited storage on the receiving end, a too short connection period, or packet prioritizing negotiated by the partners. Generally speaking, the smaller the idle period of a packet, the better the expected QoE. Again the service must decide if it can tolerate packet idle periods above a certain threshold.

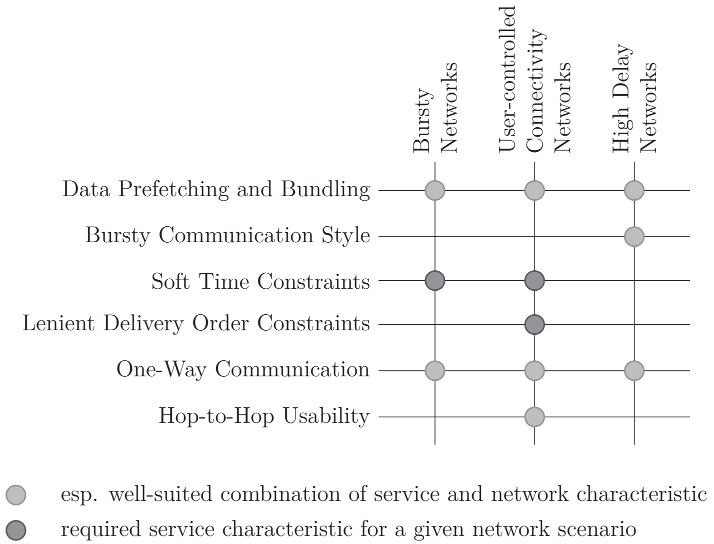


Figure 3.7 Mapping between network and service characteristics

3.7 Evaluation of DTN Services

With the properties and categories defined above, we now give guidelines on how an evaluation of network services over challenged networks should be conducted. Before starting implementation efforts, a service in question is to be analyzed of being suitable for a certain network and use case. For this, the criteria list in Section 3.2 provides guidelines on service characteristics that are indicators for good and bad suitability from a fundamental point of view. The identified characteristics shall then be matched with assumptions regarding the expected network infrastructure.

3.7.1 Unspecified More General Scenarios

While most services are designed with a specific network in mind, it should be considered that the broader the audience, the more connection variants have to be taken into account. Combining service and network characteristics lead to a *general scenario*. This scenario already allows general assumptions over the maximum QoE to expect.

Figure 3.7 shows a possible mapping between network and service characteristics. Certain service properties such as data prefetching and bundling or one-way communication improve QoE for most or all networks. Other characteristics such as hop-to-hop usability are beneficial for particular networks only, in this case: user-controlled connectivity networks. A bursty communication style with intervals of minutes between data transport needs will certainly be more efficient if used via a network with pauses between connectivity windows in the magnitude of minutes rather than hours or days.

The figure also identifies combinations that appear to be mandatory for a service if it is supposed to be suited for such a network. Soft time constraints for example seem to be a necessary prerequisite if a service is to be used via high delay networks. Regardless of the affiliation to a certain class, a network might combine properties of different classes.

3.7.2 Specific Scenarios

After deriving positive predictions from the generalized analysis, the scenario should be refined to a *specific scenario*, including predictions and assumptions about the target audience. In contrast to network and service characteristics where only a subset of categories might be applicable, all user classes presented in Section 3.4 need to be considered. The larger the targeted audience of the service, the more difficult it gets to predict involved user classes and their characteristics.

Still, a service evaluation should consider at least one main audience group. QoE metrics include both, network- and user-related indicators as well as user-only or network-only values. Regardless of the metrics being dependent on user action or reaction, any of these values will influence the users' perceived service quality. For every user group, different characteristics and metrics, as described in Section 3.6, can be important or relevant. As stated before, the nature of the service will also influence the selection of suitable QoE metrics and therefore no general recommendation of suitable metrics can be given. It should be stated though that user-involved metrics such as the users' comfort level should be considered relevant in most cases.

3.7.3 Prediction Analysis

After a first prototype is deployed, the metrics identified as relevant should be measured and the results should be compared to the former predictions resulting from the specific scenario. Hints are given in the previous section on how to measure given metrics. Comparing predictions with actual measurements should lead to either validation or refinement of assumptions regarding user characteristics or network differentiation. Finally, such a comparison give guidelines on when to declare deployments *successful* and offer opportunities for future work and improvements of the service. The presented evaluation strategy will ensure that the focus on the use case is not lost during the design phase of the service.

3.8 Related Work

To the best of our knowledge, there is no directly related work about QoE in DTNs. However, several QoS improvement approaches have been considered for DTN scenarios [Dem08]. New metrics such as delivery probability and fairness indicators where proposed [ASCS11]. Al-Siyabi et al. presented an admission control model, in which the access to a nodes resources is limited [ASCS10]. Despite that all these approaches are focused on enhancing the network's QoS in some way, they do not guarantee that the overall user experience will improve. Since the relationship between QoS and QoE is not straightforward, further research regarding QoE is required.

Routing in DTNs attracted many researchers [Zha06]. A common approach is to analyze a possible social structure of the underlying network [DH07; HCY08]. Lindgren et al. presented a routing scheme called PRoPHET for intermittently connected networks. In their approach, each node keeps a table with probabilities for reaching every possible destination in the network [LDS03]. We present our own approaches in Chapters 5 and 6.

Concerning network services, differentiation techniques have been proposed before [IP09]. Such techniques improve the delivery rate and timing based on the priority of the data request. While a couple of services have been implemented [LD07; Lin11], only few efforts even consider evaluating them from a user's perspective.

3.9 Conclusion

Traditional QoS evaluation of network services via challenged networks is not sufficient since it involves only a subset of relevant factors. To allow for more suitable evaluation efforts of upcoming work in the DTN community, we proposed new evaluation metrics based on the user's satisfaction, the Quality of Experience. Before an existing service can be considered for being used in a challenged network, it has to be analyzed if and to what extend the nature of the service permits the usage in such a scenario. To aid in this task, we identified suitable service characteristics in Section 3.2. After that, the average use case must be taken into account to an extent that at least includes the available network infrastructure. In Section 3.3 we identified network classes with respect to relevant QoS criteria. Finally, the user's specific situation must be considered, as described in Section 3.4. To put it all together, we give guidelines on how to combine these aspects and draw conclusions regarding the users perceived Quality of Experience in Section 3.7.

3.10 Acknowledgements

This work was a collaboration between André Goliath, Mónica Alejandra Lora and Jó Ágila Bitsch. Most of the implementation was performed by André Goliath in the context of his Diploma thesis [Gol12] on a new architecture for disruption aware applications.

4

Neighbor Discovery

Detecting the presence of communication partners in communication range is fundamental to intermittently connected networks, such as disruption and delay tolerant networks. In real world deployments, applications spend the majority of their operating time looking for neighbors.

As an example, consider the case of the DTN test in Laponia by Farrell et al. [FMM+11], where over the course of 60 days, between 27 nodes, 9575 contacts were recorded, averaging to one contact every 4 h. While contacts are not distributed evenly, this already hints at the time spent idly looking for communication partners.

Similarly, in sensor networks used for the observation zebras in the wild [LSZM04], sensor nodes spend the majority of the time without connectivity. Thus, reducing the energy needed during this time leads to significant energy savings.

In our RatPack project, we performed experiments of underground signal propagation of IEEE 802.15.4 and CC1000 based sensor nodes, see Section 2.4.7. These experiments further support the need for a very energy efficient discovery mechanism, as communication opportunities are rare and short.

In this chapter, we show the feasibility of perfect difference sets. This scheme results in a mathematically provable optimal schedule for neighbor discovery for two communication partners that are using the same duty cycle. Then, we consider the case, that two nodes with different energy levels are using different duty cycles and generalize our approach to the asymmetric case.

There is no general framework for energy efficient neighbor discovery yet for wireless sensor networks and Linux devices. Therefore, we present a TinyOS framework, as well as a Linux kernel module and implementation for use in applications in scenarios facing sporadic communication.

When designing the format of the discovery protocol itself, we focus on energy efficient local data sharing applications, such as a podcast sharing application. In this context we also discuss, how our system deals with neighbors which do not have new information available and can be ignored.

4.1 Contributions

Our main contributions are:

1. **Energy efficient neighbor discovery based on PDS:** We utilize Perfect Difference Sets (PDS) to discover the presence of a neighbor at the provably smallest energy cost for a given latency.
2. **Characterization of asymmetric neighbor discovery using PDS:** As different nodes are subject to different energy restrictions, we characterize the behavior of PDS based neighbor discovery when the nodes use different duty cycles.
3. **Integration of neighbor discovery into Linux and TinyOS:** We implemented an easy to use framework for Linux as well as for TinyOS to make use of neighbor discovery schemes.
4. **Energy efficient podcast sharing:** To show the feasibility of the approach for real applications, we built an energy efficient podcast sharing application on top of our Linux framework.

4.2 Existing Neighbor discovery schemes

The basis for tackling neighbor discovery mathematically is modeling it as a quorum problem. Therefore, we divide time into time slots. The challenge is then the selection of these time slots in such a way that we can guarantee mutual discovery in a defined number of slots. Quorums have been used previously in computer science. Researchers typically use them for mutual exclusion problems, see [Jou03; Mae85], but also for other fields, such as network topologies and error correcting codes.

Quorums originate from set theory. The construction of a quorum system can guarantee the intersection between subsets is not empty. An example of a very simple construction bases on a *Grid* array. Maekawa [Mae85] originally designed this scheme for mutual exclusion. The core idea is to distribute the time slots of the desired period on a rectangle. The quorum is generated by selection of an arbitrary column and row in this rectangle. Following this construction, no matter which column and row we choose, nodes are guaranteed to have at least two intersections with any other node in the network, see Figure 4.1.

Lang and Mao presented a new construction based on distributing the time slots on a *Torus* [LM98] instead of on a square. Instead of choosing a row and a column, we choose a column and a diagonal branch which is half the length of the perimeter of the torus. Making use of the wrap-around property of the torus, we can guarantee that for two nodes at least one branch intersects with the column of the other node. Instead of two intersections, we now have only one intersection between slots in most cases. *E-Torus* [JTHL05a] is an extension to this concept, adding additional branches to create an arbitrary number of guaranteed intersections.

In 2008, Dutta and Culler presented *Disco* [DC08a]. In contrast to the previous discovery schemes, which were based on geometric constructions, this scheme is

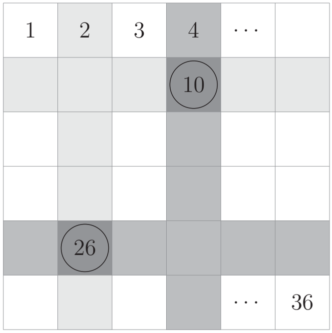


Figure 4.1 The grid quorum scheme works by arranging time slots in a rectangular array. Each node selects an arbitrary column and row. If the time slot falls into either, it turns on its radio. By construction, we can guarantee at least two intersections per period, even if nodes are not synchronized.

rooted in number theory. Each node chooses two primes. The node then turns on its transceiver during time slots which are multiples of these primes. Using a proof based on the Chinese remainder theorem, discovery can be guaranteed.

Then in 2010, Khandalu presented *U-Connect* [KLR10], a similar number theory based approach. Each node chooses a single prime number p. Nodes are active at each multiple of that prime. When two nodes choose the same prime, an intersection cannot be guaranteed. Therefore, each node performs a hyper-cycle at the square of the chosen prime. This means it sends out beacons for $\lceil p/2 \rceil$ slots. Following the Chinese Remainder Theorem, we can guarantee discovery.

As an example of a probabilistic approach, McGlynn introduced the *Birthday* protocol [MB01]. A node randomly chooses if it turns on its radio or not in each slot using weighted probabilities. Making use of the birthday paradox, we can give probabilistic guarantees that a node meets a neighbor with very high probability.

In comparison, Bluetooth device discovery [Blu10] and WiFi IBSS discovery [IEE07] do not make use of any scheduled radio off-times. Thereby, energy-efficiency is comparatively limited.

4.3 Symmetric Neighbor Discovery

In this section, we assume that each node has the same energy requirements and characteristics. Each node will therefore employ the same schedule with respect to activity of its transceiver. We will begin by formalizing the problem in a theoretic formulation. Then, we show how PDS provides a provably optimal solution for symmetric neighbor discovery.

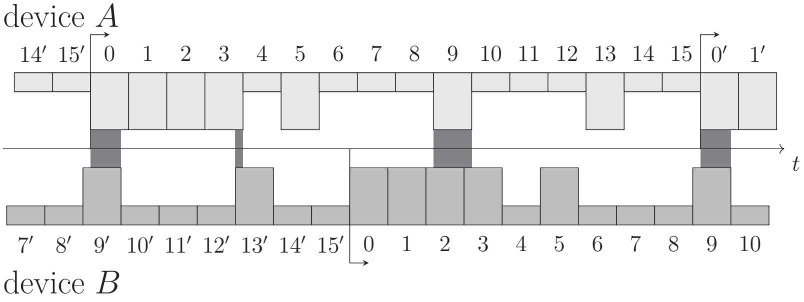


Figure 4.2 Rotation closure ensures that no matter by how far time shifts between two nodes, an intersection in the active slots still can be guaranteed. Here, two nodes using the *Grid* algorithm with a period length of $|S| = 16$ time units an offset $\phi_{i,j} = 6.8$ time units and still obtain the desired intersection.

4.3.1 Theoretic Formulation

We assume time is divided into time slots. Furthermore, we assume discovery is periodical, we therefore have a set S of $|S|$ slots. A quorum to be used for neighbor discovery has to fulfill the following four properties: (1) Non-empty intersection, (2) non-triviality, (3) equal size, and (4) rotation closure.

Definition 4.1. *Each subset of S in the quorum system needs to have a* non-empty intersection *with each other subset. If there was an empty intersection, we could not guarantee discovery.*

$$\forall S_i, S_j \in S, i \neq j : S_i \cap S_j \neq \emptyset$$

Definition 4.2. *We require quorums to be* non-trivial*, so no subset of S can be a superset of the other. The only quorum system with one subset being a superset of another would be the trivial quorum system, with the subset being identical to S.*

$$\forall S_i, S_j \in S, i \neq j : S_i \not\subset S_j$$

Definition 4.3. *As each node is assumed to have the same energy requirements, we require each used subset to be of* equal size*.*

$$\forall S_i, S_j \in S, i \neq j : |S_i| = |S_j|$$

Definition 4.4. *Finally, we require* rotation closure *(cf. Figure 4.2). For any $S_i, S_j, i \neq j$ with $\phi_{i,j}$ being the phase offset between i and j and $\Psi(i,t)$ being the neighbor discovery schedule of node i at time t with $\Psi(i,t) = 1$ if i is active at time t:*

$$\forall t \exists \phi_{i,j} : \Psi(i,t) = \Psi(j, t + \phi_{i,j}) = 1$$

This last requirement has two consequences: (1) Each node can use the same schedule without node specific initialization. (2) Nodes are not required to be synchronized at all as an arbitrary offset still leads to a discovery.

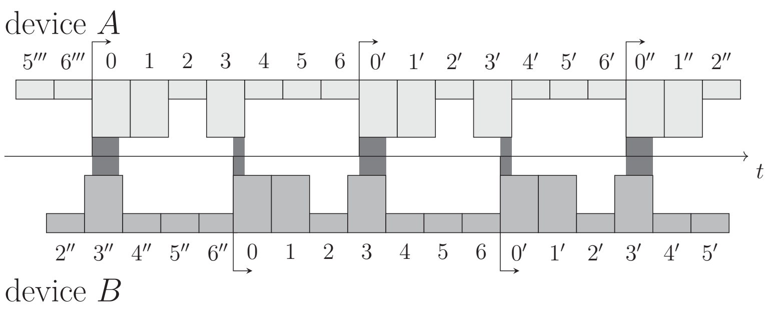


Figure 4.3 The shortest possible schedule based on the perfect difference set from $p^n = 2^1$ is $N = p^{2n} + p^n + 1 = 7$ time units long and can be represented by $G = \{0; 1; 3\}$. Through construction, the rotation closure holds, while at the same time guaranteeing a minimum duty cycle for a given latency. The example offset is $\phi_{i,j} = 3.7$.

4.3.2 Perfect Difference Sets

Maekawa [Mae85] proved the lower bound for the quorum size for mutual exclusion and showed that finite projective planes fulfill this property. Taking the requirement of equal size, he derives that a quorum for each element of S exists. He then defines D, the number of quorums in which an element of S is represented. As every element is represented in its own quorum, it must be contained in $D - 1$ quorums.

Following this, for K being the desired size of the quorum system, the number of distinct members in S must equal:

$$N = (D - 1) \cdot K + 1$$

As the number of elements in S must be equal to the union of the elements of all quorums, it follows that $K = D$.

$$N = \frac{K \cdot N}{D} \Rightarrow K = D$$

Therefore, we obtain:

$$N = (K - 1) \cdot K + 1$$

With $K = p^n + 1$, with p^n being a power of a prime, this corresponds to the number of vertices and edges in a finite projective plane, for which non-empty intersection also holds. Following the argument of Tien-Tsin et al. [TTLW97], we can therefore reduce the problem of finding the required quorum with finding the appropriate finite projective plane. Thus, the order of the size of the quorums with respect to the size of S, $N = |S|$ is:

Algorithm	power–latency–product	factor of $\sqrt{N}$	2%	5%	10%	20%
Disco	$2 \cdot \sqrt{p_i p_j}$	≈ 2	27719	3473	1199	119
E-Torus	$\sqrt{N} + \left\lceil \frac{\sqrt{N}}{2} \right\rceil \cdot B$	$\approx 1 + B/2$	15625	2500	900	100
Grid	$2 \cdot \sqrt{N} - 1$	≈ 2	9801	1521	361	81
PDS	$p^n + 1$	≈ 1	2451	381	91	31
Torus	$\sqrt{N} + \left\lceil \frac{\sqrt{N}}{2} \right\rceil$	≈ 1.5	5625	900	225	49
U-Connect	$p + \left\lceil \frac{p}{2} \right\rceil$	≈ 1.5	5366	856	176	51

Table 4.1 We show the power latency product for different algorithms and how long a period takes for a given duty cycle. PDS allows for the shortest schedules. For E-Torus, we chose $B = 2$.

$$
\begin{aligned}
& & N &= p^{2n} + p^n + 1 \\
& & &= p^n(p^n + 1) + 1 \\
\Rightarrow & & (p^n + 1)^2 - p^n &< (p^n + 1)^2 \\
\Rightarrow & & & \text{size of quorums } \mathcal{O}(\sqrt{N})
\end{aligned}
$$

Thus the size of the resulting quorums meets the lower bound $\mathcal{O}(\sqrt{N})$.

For the actual construction of a discovery schedule, we use the result of Singer [Sin38a; CD06]. He constructs *perfect difference sets* from finite projective planes. These are numberings of the nodes in finite projective planes, that allow us to create all elements 0 to $N - 1$ as the difference between two elements of this set.

While perfect difference sets originally have a different interpretation, they perfectly fit the quorum properties needed for neighbor discovery. For a p^n, the power of a prime, we can create quorums of size $p^n + 1$ to cover $p^{2n} + p^n + 1$ slots.

As we can create all differences between 0 to $N - 1$ for $N = p^{2n} + p^n + 1$, it follows directly that we have rotation closure. All quorums have the same size, as they are the same schedule rotated. The quorums are not trivial and there is an intersection between all quorums. Therefore, the slot schedule is exactly the elements from the perfect difference sets.

To give an example, consider $p^n = 2^1 = 2$. This results in the perfect difference set $G = \{0; 1; 3\}$ for covering schedules of length $p^{2n} + p^n + 1 = 7$, which is the period where the schedule repeats. Therefore, the complete schedule is $\Psi_i = (1, 1, 0, 1, 0, 0, 0)$. This schedule has a duty cycle of $3/7 \approx 43\,\%$, cf. Figure 4.3.

In Table 4.1, we show the power latency product for the different neighbor discovery schemes and associated period lengths for a given duty cycle. The duty cycle for PDS is the number of active slots divided by the number of all slots:

$$DC = \frac{K}{N} = \frac{p^n + 1}{p^{2n} + p^n + 1}$$

Accordingly, the power–latency–product for PDS is:

$$PLP = DC \cdot N = \frac{K}{N} \cdot N = K = p^n + 1 \approx \sqrt{N}$$

The power–latency–product of Torus ($\sqrt{N} + \left\lceil \frac{\sqrt{N}}{2} \right\rceil \approx 1.5 \cdot \sqrt{N}$), E-Torus ($\sqrt{N} + \left\lceil \frac{\sqrt{N}}{2} \right\rceil \cdot B$) and Grid ($2 \cdot \sqrt{N} - 1$) follow from their construction. The values for Disco ($2 \cdot \sqrt{p_i p_j} = 2 \cdot \sqrt{N}$) and U-Connect ($p + \left\lceil \frac{p}{2} \right\rceil \approx 1.5 \cdot \sqrt{N}$) are taken from [KLR10]. We notice, that PDS is the only algorithm that provides a 1–approximation of $\sqrt{N}$.

4.4 Asymmetric Neighbor Discovery

In the context of perpetually running networks, nodes may have widely differing energy budgets. Consider TurtleNet [Sor09], where researchers equipped turtles with sensor nodes. Those nodes also include solar panels to harvest energy. As a subset of turtles are exposed to the sun more often than others, their respective energy budget is higher. Other nodes will have to use a more restrictive duty cycle leading to asymmetric duty cycles.

The same reasoning applies to heterogeneous mobile devices with different battery sizes. As the battery size determines the maximum time a mobile device may stay active in the absence of energy harvesting or recharging opportunities, devices with smaller batteries have a lower energy budget available during times of energy scarcity.

In this section, we discuss fairness in this generalized, asymmetric case. We then characterize how *PDS* deals with this situation and how *Grid* as an analytically more tractable approach handles asymmetry.

4.4.1 Fairness in Neighbor Discovery

Consider the following scenario: Node A and C harvest 10 times more energy than node B. The latency of a neighbor discovery protocol will depend on the node B, the least active one. Since nodes A and C can spend more energy on neighbor discovery, they should not be penalized with a high latency but rather benefit from a resulting smaller latency. We capture this intuitive notion in a more formal way by defining the *Energy Gap*.

Definition 4.5. *Given two nodes A and B, that can harvest energy e_A and e_B respectively. Let $e_A, e_B > 0$ and $e_A \geq e_B$. The* Energy Gap *between nodes A and B is defined to be the ratio:*

$$\alpha_{A,B} = \frac{e_B}{e_A}$$

Given the above definition, we can now formalize our *Fairness* criteria. If the energy gap $\alpha_{A,B}$ between node A and B is closer to 1, the average latency $\bar{l}_{A,B}$ must be smaller compared to when the energy gap $\alpha_{A,B}$ is closer to 0.

Definition 4.6. *So, an algorithm is* fair*, if the following holds:*

$$\forall B, C | \alpha_{A,B} \geq \alpha_{A,C} : \bar{l}_{A,B} \leq \bar{l}_{A,C}$$

4.4.2 Fairness of Grid-based Neighbor Discovery

In the following, we will prove that the *Grid* discovery scheme is indeed fair according to this definition.

Proof. For simplicity without loss of generality, we consider the *Grid* neighbor discovery scheme on a $n \times n$ grid. The algorithm chooses one row and one column to be active in.

Therefore, there are $2n - 1$ active slots out of a total of n^2 slots. Thus, the duty cycle is:

$$\frac{2n-1}{n^2} \approx \frac{2}{n}$$

Accordingly, a more energy constrained node chooses a greater n.

If two nodes N and M choose values n and m, with $n \leq m$, the energy gap is $\alpha_{N,M} \approx \frac{2}{m}/\frac{2}{n} = \frac{n}{m}$. Thus, we must show, that for two values m_1 and m_2, with $m_1 \leq m_2$, the average latency $\bar{l}_{N,M_1}$ is smaller than the average latency $\bar{l}_{N,M_2}$. In the following, we will see that this is possible.

Looking at the actual schedule of the *Grid* scheme, we see that there are two types of active periods. A long continuous period of length n for the chosen row and $n - 1$ single slots each separated by $n - 1$ inactive slots for the chosen column, see Figure 4.1. For two nodes N and M, with two different parameters n and m where $n \leq m$ we can guarantee that one of the single slots of node N will always intersect with the long slot of node M because the long slot length is larger than the inter-slot spacing of the short slots by construction.

If we assume an equal distribution of offsets, we are on average $\frac{m \cdot m}{2}$ away from said intersection, the worst case being m^2. This average case (as well as the worst case) only depend on the more energy constrained node M. From this follows that *Grid Neighbor Discovery* is indeed fair for asymmetric neighbor discovery, as defined above. □

4.4.3 Fairness of Perfect Difference Sets-based Discovery

Unfortunately, the internal structure of a perfect difference set does not exhibit any usable regularity, we can use to guarantee average discovery latency. Indeed, for specific rare offset and schedule combinations, where the schedule lengths are not coprime, there exist wrap arounds, where no intersection can be found. Examples of this are the schedules for the combination of schedule lengths $\{91; 273\}$ and $\{651; 4557\}$. We did an exhaustive search, and those where the only combinations where nodes have a duty cycle $\geq 1\,\%$. In practice, however, perfect difference sets behave fair, as we will discuss in Section 4.6.3.

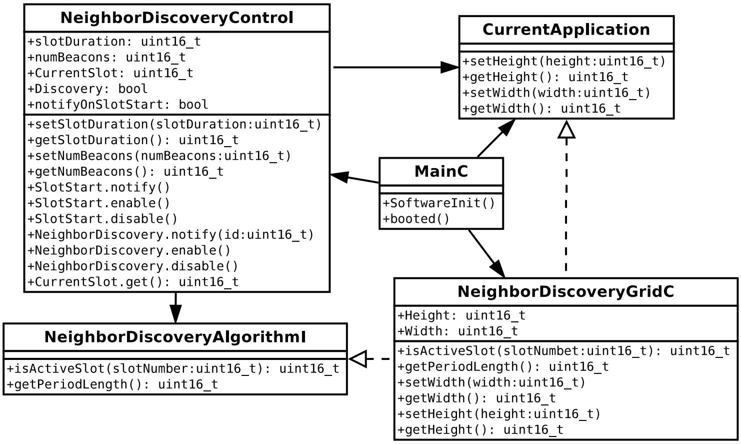


Figure 4.4 Communication between modules is handled via the NeighborDiscoveryAlgorithmI interface. This allows us to easily replace the Discovery Scheme with minimal changes to the surrounding program and handling. In this example, we use grid based discovery.

4.5 Implementation of Neighbor Discovery Schemes

We implemented the different neighbor discovery schemes for three different platforms. Initially, we created (1) a Python simulation to investigate the properties of different schemes on a high level. Driven by the need of DTN mechanisms in selected sensor network applications like RatPack, cf. Section 2.4.7, (2) we implemented a TinyOS library, which we used also for evaluation in the Motelab [Har05] testbed. Finally, we also implemented the different neighbor discovery schemes for (3) the current Linux kernel and build a energy efficient file sharing application. In the following, we discuss our practical considerations and abstractions.

4.5.1 Simulations

Although, our initial use case is sensor networks, we decided to first implement a very abstract Python simulation. This allowed us to better investigate different ideas with respect to perfect difference sets, as well as fairness. Not having to consider environmental factors, such as link loss and radio mode switching times, we are able to perform exhaustive searches, as in Section 4.6.3, and gather more statistics.

4.5.2 TinyOS

One of the most popular sensor network operating systems at the time of writing is TinyOS [Pil10]. It is organized in a very modular way, and has a variety of backends. Using the common `ActiveMessageC` abstraction, we built a Neighbor Discovery Framework that is instantly portable to all sensor network platforms, TinyOS supports. We organized the structure of our modules in such a way that

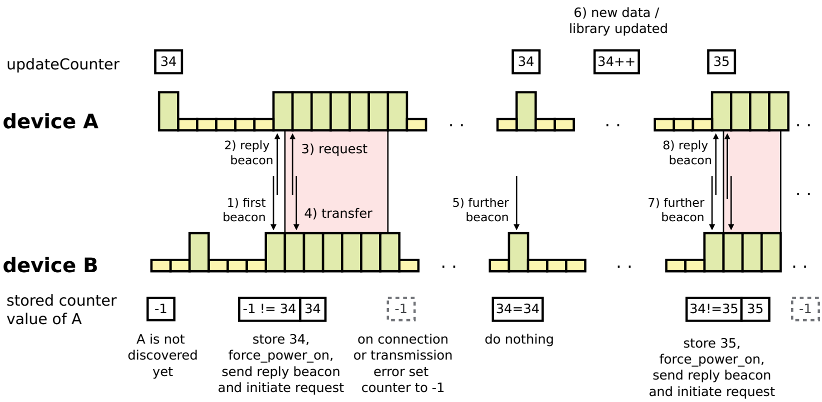


Figure 4.5 When a node is discovered, an initial exchange takes place. From this point on, the node stores an update counter. The update counter changes as the system state of a node changes, e.g. when new data becomes available. Only when this update counter changes, a neighboring node will connect to the node.

the neighbor discovery algorithm is exchangeable very easily, cf. Figure 4.4. For our evaluation, we also implemented all the algorithms presented in the related work.

4.5.3 Wi-Fi

We make use of generic callbacks in the Linux Wireless `mac80211` kernel module to the wireless hardware device's driver. These callbacks, named IEEE80211 operations, allow us to stay independent of the different specific hardware implementations. We implement one additional method into the `mac80211` module to control the *start* and *stop* callback functions and export it to kernel space. In addition, we built a new kernel module, `LiND_pwr`, short for **Li**nux **N**eighbor **D**iscovery **pow**er control, which uses the exported method and provides a character device to user space.

Thus, applications can control the power of the Wi-Fi device's transmitter by writing to `/dev/LiND_pwr`. Our design allows an easy exchange of the neighbor discovery application, which implements the specific neighbor discovery algorithm. Beacons are injected and captured by setting the operation state of an additional virtual network interface to monitor-mode. Our beacons include an integer value, called `updateCounter`. This value serves as an indicator in order to determine if it is useful to contact a previously discovered node again as its internal state changed. A request is initiated by sending a reply beacon, which would inform the intended receiver to stay continuously active during the exchange of additional data over TCP/IP, see Figure 4.5.

4.6 Experimental Results

Aside from initial experiments and the exhaustive search for asymmetric discovery latency for PDS, all experiments were performed on real nodes. The sensor network

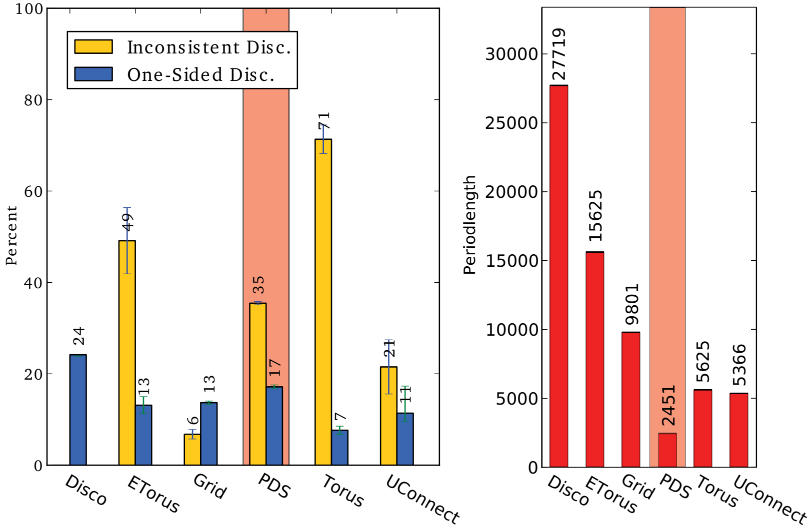


Figure 4.6 Results of tests on Motelab with an applied duty cycle of 2% to every algorithm. Inconsistent and one-sided discovery errors (left). Period lengths in number of slots (right).

experiments were conducted using Motelab [Har05]. We used a fixed slot length of 100 ms for the Motelab and for the Wi-Fi tests.

4.6.1 Motelab

The tests were run for half an hour for each algorithm, as this is the maximum continuous time slot, for which we could continuously access the Motelab testbed before it schedules a new test. This is a restriction of Motelab on which we had no influence. The analyzed duty cycles were 2%, 5%, 10% and 20%.

We used two metrics to measure the quality of an algorithm: (1) *Inconsistent Discoveries* and (2) *One-Sided Discoveries*. Inconsistent Discoveries measures when for a fixed node, another node which is known to be in radio range is not detected within a discovery period. This captures how often a discovery fails. A one-sided discovery happens, when one node discovers the presence of the other, but not vice-versa, therefore, the asymmetry in the discovery.

As seen in Figure 4.6, for a duty cycle of 2% only PDS has a sufficiently short period length of 245.1 s $\sim$ 4 min. This coincides with our analytical predictions in Table 4.1. In general, for algorithms that produce long periods of consecutive active slots, such as *Grid*, the above-mentioned quality metrics are better. This coincides with the requirement of less radio mode switching, which might lead to better results for higher duty cycles. Keep in mind, there is no coordination necessary between the nodes in any way, see Section 4.3.1.

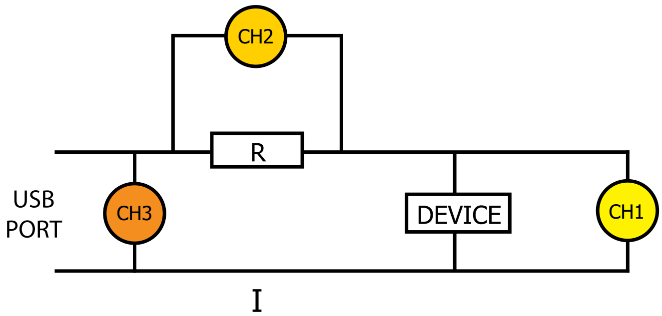


Figure 4.7 Circuit diagram of our experimental setup. We performed our measurements using a wireless network card connected via USB. The reason of this is, that this allowed us to measure the required energy through inserting an additional measurement resistor ($0.108\,\Omega$) in the USB cable. Measurement electrodes record the different voltages V_{CH1}, V_{CH2} and V_{CH3} for a later calculation of the power consumption of our wireless USB device. Oscilloscope type: Tektronix, TDS 2024B (200MHz, 2GS/s).

4.6.2 WiFi Tests

Switching a WiFi device's radio transmitter *off* or *on* takes a specific period of time and consumes additional power, which varies between different hardware models and vendors. To toggle our test device, a *TL-WN422G* USB adapter, from *off* to *on* or vice-versa, constantly takes 12 ms. During one complete switch, from *off* to *on* and back to *off* again, the device is additionally powered for 14 ms of the overall period of 24 ms.

However, using a neighbor discovery scheme like *U-Connect* with a maximum discovery latency set to 10 s, we achieved discovery within an average latency time of 3.2 s, while the WiFi transmitter is powered only 17% of the time. The average energy consumption of our wireless device during an *on* state is 90 mW higher than in an *off* state. For our test set-up, see Figure 4.7. Visualizations of our power measurements, see Figure 4.8 and Figure 4.9 illustrate the significant energy savings of the application of our `LiND_pwr` kernel module compared to continuous *Ad-hoc mode*.

The USB interface in our test setup has a a baseline power consumption of 0.18 W. The continuous power consumption of ad hoc mot is 0.28 W, while our approach only has this consumption in the short intervals were the radio needs to be active according to the discovery schedule. We expect, that a System on Chip device as common in current mobile devices can lead to a significant additional reduction in power draw, as the USB subsystem will not have to be continuously powered.

4.6.3 Asymmetric Discovery of Perfect Difference Sets

Using the Chinese remainder theorem, we can prove that while schedules intersect with themselves by construction, different schedules intersect, if $p_1^{2n_1} + p_1^{n_1} + 1$ is co-prime to $p_2^{2n_2} + p_2^{n_2} + 1$ at least at a cycle length of $\left(p_1^{2n_1} + p_1^{n_1} + 1\right) \cdot \left(p_2^{2n_2} + p_2^{n_2} + 1\right)$. While the structure of perfect difference sets does not lend itself to further formal

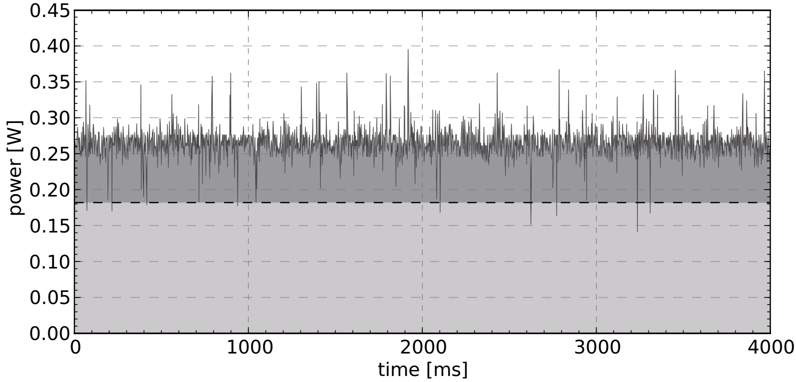


Figure 4.8 This diagram shows the overall power consumption of the wireless device running in ad hoc mode. Please note the baseline consumption of about $0.18\,\mathrm{W}$, and the continuous consumption of about $0.28\,\mathrm{W}$.

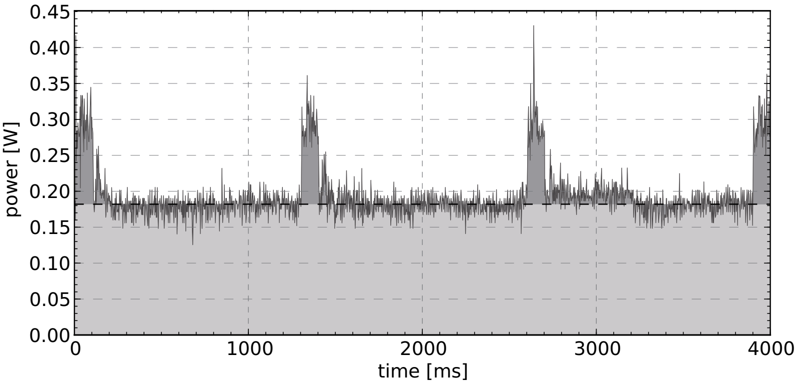


Figure 4.9 This diagram shows the power consumption of the ad hoc mode in combination with `LiND_pwr` using the *U-Connect* neighbor discovery algorithm. Note that the power consumption returns to the baseline, with only short peaks of activity in between. A wireless network interface directly integrated into a device will likely have a considerably lower baseline.

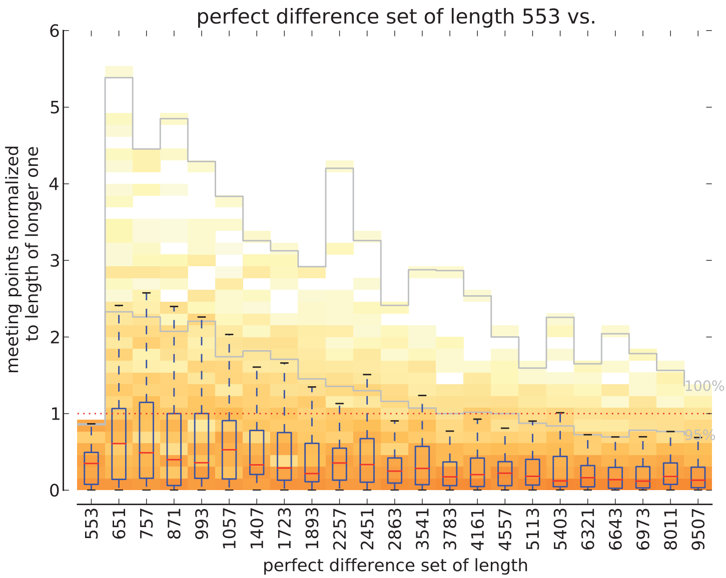


Figure 4.10 We compare the PDS of length 553 to larger PDSs. While we normalize the y-axis to the length of the longer PDS, we notice in the boxplots, that even though the worst case performance, see the 100 percentile line, is not favorable, the large majority of discoveries happens within the length of the longer PDS, see also the 95 percentile line. The background shows the density of discoveries with a particular delay, assuming an equal distribution of all possible offsets between schedules. Discoveries actually happen faster than with *Grid*, especially as we consider longer PDSs.

guarantees, the full enumeration of all possible practical schedules ($DC \geq 1\%$) and offsets shows that in practice, perfect difference sets behave fairly.

Figure 4.10 shows the comparison of a perfect different set of length 553 ($p^n = 23^1$, $DC = {}^{24}/_{553} \approx 4.3\%$) to larger perfect difference sets (that is even smaller duty cycles, down to $DC \approx 1\%$). We therefore showed that perfect difference sets are useable and beneficial approach, not only in theory but also in practice for symmetrical and asymmetrical asynchronous neighbor discovery.

4.7 Conclusions

We presented an energy efficient neighbor discovery scheme based on perfect difference sets with the provably optimal power-latency trade-off, which does not require any coordination between nodes. We defined fairness for asymmetric neighbor discovery, proved the *Grid* scheme to be fair, and showed that our scheme behaves fair for duty-cycles used in practice. We implemented a neighbor discovery framework into Linux and TinyOS for general use. Finally, we used our framework to build a podcast sharing application. In this context, we show how to deal with nodes which are present but do not have new available data and can be ignored.

4.8 Acknowledgments

We first looked at the neighbor discovery problem in more detail during the Master thesis of Qian Wu [Wu09]. We developed the theoretical foundation of the Perfect Difference Set approach in the Bachelor thesis of Stefan Schupp [Sch10b]. The concept of fairness for asymmetric discovery came up in discussions with Ragesh Jaiswal and Amitabha Bagchi of IIT, Delhi during a scientific stay in New Delhi, India, in the summer of 2010. Mr Christoph Wollgarten helped implementing the Linux kernel module in his Diploma thesis [Wol11]. We published a write up of this work at ExtremeCom 2011 [BLWSW11].

5

Social Network Based Forwarding

As introduced in Chapter 2, Pocket Switched Networks [HCG+06] or more generally Disruption/Delay Tolerant Networks are a representative of a new type of communication paradigm, in which data is delivered over eventually transportable dynamic networks. Data bundles are cached for a possibly long time before a suitable contact to a forwarding node is made. The proliferation of mobile devices leads us to the belief, that this paradigm will gain significant popularity in the future.

The routing of data in this kind of networks can be categorized in several classes [FC06]. In the simplest case, a data bundle is epidemically forwarded in the network. More refined approaches forward either dependent on detailed plans of future connectivity or heuristics to predict the probabilities of those. Ramanathan et al. [RBK07] provides a more formal description of DTNs.

The identification of useful heuristics is therefore the core contribution of the majority of publications about routing in these kinds of networks. The social structure has long been considered a good starting point for building heuristics in the community. However, the change over time of this structure, especially over long runs of time, has often been neglected.

The contribution of this chapter is therefore the adaption of a previous social network based approach, namely SimBet [DH07], to consider the *progression* of social networks over time. To that end, the original definitions of *similarity* and *betweenness* had to be significantly adapted to deal with weighted graphs instead of binary ones. Additionally, we described a novel metric, the *directed ego flow betweenness*.

5.1 Background and Related Work

One of the simplest approaches to use aging in MANETs is FRESH [DFGV03]. Interpreting the time of the last contact to a destination node as a measure of the distance, FRESH is able to discover short path in an efficient way. Although this approach needs virtually every node to encounter the destination at least once, it shows the importance of the age of a contact as a parameter for efficient routing.

Lindgren et al. [LDS04] presented a probabilistic routing scheme for intermittently connected networks called *PRoPHET* where each node keeps a table with probabilities for reaching every possible destination in the network. These probabilities depend on the frequency of observations, the time of absence and the distance to a destination, but does not make use of the social nature of the network.

Bubble Rap [HCY08] explicitly discovers structural features by processing historical contacts and computing node communities and hierarchies a priori. However, Bubble Rap requires a setup period and does not adapt to dynamics in the network.

Our work builds upon previous work done by Daly et al. [DH07] who were one of the first to introduce social network analysis in the context of delay tolerant networks, called *SimBet*. In this approach, routing decisions are based on two common properties of social networks, *social similarity* and *centrality betweenness*.

In contrast to PRoPHET, SimBet relies only on an ego-centric network view and is able to make efficient forwarding decisions even if the node has not encountered a path to the destination before. Their handling of node distances inspired us with respect to modeling the dynamics in relations as presented in Section 5.2.1.

The calculation of ego betweenness in graphs with weighted edges was introduced by Freeman et al. [FBW91]. We describe the approach in more detail in Section 5.2.3. We extend this concept to a directed ego flow betweenness in Section 5.2.5.

5.1.1 Background on SimBet Routing

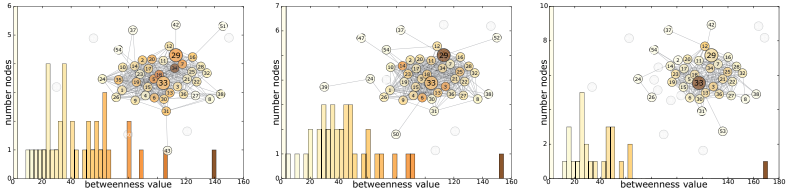
The underlying concept of SimBet is to use social network analysis to provide an efficient routing scheme that has no need for global knowledge about the network. Conceptually, a node forwards messages to a node that has a higher social similarity to the destination, i.e., it is more likely to have a contact to the destination. If there is no such node known to the forwarder, it instead forwards the message to nodes with a higher betweenness, i.e., to a potentially better forwarder to other parts and clusters in the network.

While similarity is defined from an ego-centric view, betweenness is defined on geodesics, the shortest paths between two points. Therefore, the calculation needs global knowledge. However, Everett et all [EB05] showed that the calculation of an ego-centric betweenness is still meaningful enough to fit a global representation.

The social structure can be used as the macro structure for forwarding decisions. The similarity contributes strongest when a message is already near to the destination with respect to the social structure, whereas the betweenness has stronger effects when the forwarding node is far from the destination.

Accordingly, the similarity $S_u(v)$ of u to v can be defined as the number of common encounters between u and v:

$$S_u(v) := |N_1(u) \cap N_1(v)| \tag{5.1}$$



(a) Using only the first third. (b) Using only the second third. (c) Using only the third third.

Figure 5.1 The distribution of the ego betweenness using the original SimBet algorithm processing slices of the *imote* trace. The structure changes over time. Nodes with a high betweenness in subfigure (b), e.g. node 29, have a very low betweenness at a later point in time, while others display a higher betweenness only later, e.g. node 33 in subfigure (c). The original SimBet algorithm does not reflect these changes when run on the complete data set.

The definition of the global betweenness centrality $C_B(u)$ of u depends on all geodesics going through u as a fraction of all geodesics $g_{v,w}$:

$$C_B(u) := \sum_{\substack{v \neq w \neq u \\ v,w \in \text{Nodes}}} \frac{g_{v,w}(u)}{g_{v,w}} \tag{5.2}$$

Thus, Daly defines the ego betweenness centrality by taking nodes v and w only from the neighborhood of u:

$$\theta_u := \sum_{\substack{v \neq w \neq u \\ v,w \in N_1(u)}} \frac{g_{v,w}(u)}{g_{v,w}} \tag{5.3}$$

5.1.2 Shortcomings on SimBet

Despite the fact that SimBet performs quite well as shown by the evaluations done by its authors [DH07], there are a number of weaknesses. Routing decisions in SimBet are based on comparing the social relations and roles of individual nodes at a specific time in a binary fashion. However, relations differ in strength, i.e., contacts between nodes can have different frequencies. Additionally, those relations may change over time. Therefore, a simple statement that x is in contact with y without additional qualifiers is not enough.

Further, a node might have different roles: It may participate in different clusters (different circles of friends) so one node might be important for forwarding information from one cluster to another but not necessarily that important for forwarding within a cluster (high global betweenness, but low directed betweenness and similarity).

In Figure 5.1, we see how the network structure and the betweenness of individual nodes varies over time. Comparing its three subfigures, it becomes evident that the original model based on the ego betweenness centrality in a network is not adequate and will lead to inaccurate decisions for transmissions at different times t.

SimBet's shortcomings can therefore be summarized as follows:

$G(t)$	time dependent graph
$u, v, w \in V$	vertices u, v, w
$N_i(u)$	i-hop neighborhood of u
$e \in E$	edge e
$\omega(e, t)$	time-dependent weight or *freshness* of edge e, also abbreviated ω
$\omega(P, t)$	freshness of a path P at time t
$\omega(u, v)$	current weight between nodes u and v
α	renewing/growth factor
γ	aging/decay factor
Δt	time steps since a previous contact
$g_{v,w}$	number of geodesics between v and w
$g_{v,w}(u)$	number of geodesics between v and w passing through u
$S_u(v)$	similarity between u and v
$C_B(u)$	betweenness centrality of u
$\theta(u)$	ego betweenness centrality of u
$\sigma(u, v)$	aged similarity between u and v
ϑ_u	ego flow betweenness centrality of u
$\delta_u(d)$	destination ego flow betweenness centrality of u with respect to the destination d
$U_v(d)$	utility of v for transporting a message to d

Table 5.1 List of symbols and abbreviations used in this chapter

1. Representing a social relation in a **binary form is not realistic** and will lead to inconsistencies when comparing such relations.

2. The macro structure of a social (dynamic) network is **changing over time**, i.e. nodes enter, leave or switch their neighborhood and so the social role will become a different one.

3. The betweenness value calculation is meant to consider only shortest paths, but **social communication is not limited only to the shortest possible paths**.

4. The **global betweenness utility of a single node becomes less important the closer a message comes to the destination**. If the message reaches the destination's neighborhood, destination betweenness can be more precise by taking only relevant directions into account.

In the following, we will present our approach to counter these limitations through generalizing the different graph metrics.

5.2 SimBetAge

In this section, we present SimBetAge. While building upon SimBet, we deal with the dynamics of social networks and adjust the used social network metrics significantly. Doing so, we increase performance drastically.

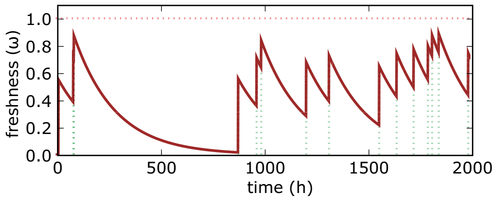


Figure 5.2 The *freshness* ω of a single edge with some encounter events over time. The higher the value, the greater the probability of encounters in the near future.

5.2.1 Modelling the dynamics in relations

A binary model of a social relation, i.e., one node knows another node or not, does not reflect the dynamics inherent to such a network. Using a weighted time dependent graph $G(t)$ as a model of a social network will bring a more realistic view, where $G(t) = (V, E, \omega(e, t))$ is a fully connected graph. This weight of an edge is what we will call the *freshness* of an edge, where $\omega(e, t) = 0, e = (u, v) \in E, t \in T$ means, that the nodes u and v have never be connected from time t_0 up to time t and $\omega(e, t) = 1$ represents a permanent connection between u and v.

The *freshness* of a single contact is influenced by two events: The first one is a time step event that will decrease the freshness value, i.e. the contact becomes older, which is modeled through the use of an exponential decay function. The second one is an encounter event that will increase the freshness value, i.e. the contact becomes *fresher* which is modeled by a logistic growth function. Figure 5.2 shows the exemplary development of a freshness value over time. Mathematically, we model these events according to Equation 5.4 and Equation 5.5:

$$\omega_{\text{new}} = \omega_{\text{old}} \cdot \gamma^{\Delta t} \tag{5.4}$$

$$\omega_{\text{new}} = \omega_{\text{old}} + (1 - \omega_{\text{old}}) \cdot \alpha \tag{5.5}$$

After we have defined the freshness of a single edge, we also need to define the freshness of a path P with length $l(P) > 1$. Due to the representation by logistic growth and exponential decay for the age value of a single edge, the freshness $\omega(e, t)$ can also be interpreted as an indicator, how probable it is that the two nodes u and v are connected at time t. Hence, the freshness $\omega(P, t)$ of a path P is defined by the product of all freshness values in P:

$$\omega\left(P, t\right) := \prod_{e \in P} \omega\left(e, t\right) \tag{5.6}$$

Given these definitions for aged edges and aged paths, we now define our calculation for similarity and betweenness in aged time-varying graphs.

5.2.2 Similarity for Aged Graphs

In a binary graph the similarity $S_{\text{binary}}(u, v) = |N(u) \cap N(v)|$ of two nodes u and v is defined by the number of common neighbors between u and v. In a graph with aged relations, we define the similarity of two nodes as being proportional to the freshness of their shared neighborhood. If node u and node v share a neighbor w with the freshness values $\omega(u, w)$ and $\omega(v, w)$, then the similarity $\sigma_w(u, v)$ of node u and v with respect to neighbor w is asymptotically equal to the product of the freshness values of their relations to w.

The calculation of the similarity is based on the encounter information exchanged during runtime. Therefore, we need to consider that the encounter information at node u about node v might be outdated proportionally to the time node u has not heard about the encounters of node v with w either from node v or from node w. This means, the similarity of node u and node v concerning w needs to be scaled by the freshness of the encounter information of node v about the common neighbor w.

$$\sigma_w(u, v) := (\text{scale for freshness}) \cdot (\text{strength } u \text{ to } w) \cdot (\text{strength } v \text{ to } w)$$

$$\sigma_w(u, v) := \begin{cases} \omega_{u,v} \cdot \omega_{u,w} \cdot \omega_{v,w} & \text{if } v \in N_1(u) \\ \omega_{u,w} \cdot \omega_{u,w} \cdot \omega_{v,w} & \text{otherwise} \end{cases} \tag{5.7}$$

Following this definition, we then define the aged similarity $\sigma(u, v)$ as the sum of the similarities of all common neighbors w between node u and v.

$$\sigma(u, v) := \sum_{w \in N_1(u) \cap N_1(v)} \sigma_w(u, v) \tag{5.8}$$

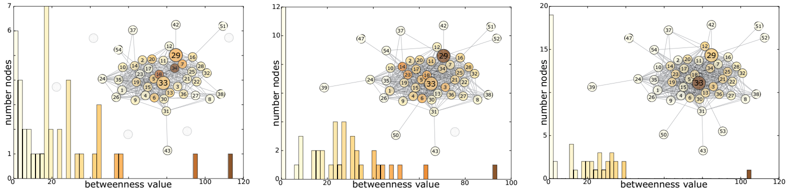
5.2.3 Betweenness for Aged Graphs

Analogous to the calculation of the betweenness in binary graphs, in weighted graphs only geodesics between two nodes are taken into account. In an aged graph, those are the paths with the highest freshness value $\omega(P, t)$. The method for calculating stays the same as the binary betweenness as in Equation 5.2. However, the geodesics may change suddenly and significantly depending on the individual freshness values, while for routing decisions, smoother functions are preferable since they better model the connectivity in social networks.

5.2.4 Flow Betweenness

When looking at communication characteristics in social networks, the question arises why only shortest paths linking two individuals should be the possible way of communication flow. In a stochastic network model, you might be advantageous to use all possible paths between two nodes for a more representative view.

According to this model, [FBW91] defines the flow betweenness. Instead of the number of geodesics, the calculation takes all possible paths in a network into



(a) After the first third. (b) After two thirds. (c) After the whole trace.

Figure 5.3 Snapshots of the distribution of the ego betweenness using our algorithm for betweenness for aged graphs described in Section 5.2.3 in the *imote* trace. It closely follows the change in structure in Figure 5.1, e.g. in (b) Node 29 has a high betweenness value, while in (c) Node 33 has the highest betweenness value.

account, which was inspired by [SZ89]. As a result of their observations, they came up with a centrality measure more consistent and susceptible to dynamic network processes than other centrality measures.

Flow betweenness centrality is based on Ford and Fulkerson's [FF56] model of network flows. We therefore define the ego-centric flow betweenness value ϑ_u as the sum over the freshness of all paths between pairs of neighbors (v, w) of u passing through u divided by the freshness of all possible paths between them, weighted with the freshness of the edge between u and v and the one between u and w:

$$\vartheta_u := \sum_{v,w \in \text{neighborhood}(u)} \frac{(\text{scale for freshness}) \cdot (\text{strength of path } v \text{ to } u \text{ to } w)}{(\text{direct path } u \text{ to } v) + (\text{all paths } u \text{ to } x \text{ to } v \text{ for all } x)}$$

$$\vartheta_u := \sum_{v,w \in N_1(u)} \frac{(\omega_{u,v} \cdot \omega_{u,w})^2}{\omega_{u,v} + \sum\limits_{\substack{x \in \{u\} \cup N_1(u) \\ x \neq v \neq w}} \omega_{x,v} \cdot \omega_{x,w}} \tag{5.9}$$

Figure 5.3 demonstrates the influences of aging on the flow betweenness distribution of a network evolving over time. Comparing it with Figure 5.1, we see that the distribution follows the dynamics and structural changes in the network now very closely, thus making the representation more consistent with the real world.

5.2.5 Directed Betweenness

In an ego-centric view, we distinguish between three decision states when two nodes u and v compare their utilities for a destination d:

1. Both nodes are far away from the destination:
   $d \notin N_1(u) \cup N_1(v)$
2. At least one node has the destination in its two hop but not one hop proximity:
   $d \in (N_2(u) \cup N_2(v)) \setminus (N_1(u) \cup N_1(v))$

3. At least one node is close to the destination:
   $d \in N_1(u) \cup N_1(v)$

In the first case, the most general value, the betweenness value, is a sensible indicator, as the most between node is the most central to a network. In the third case, the higher similarity will be the best choice. However, the second case might be critical as the similarity might not be meaningful enough and a high betweenness could possibly lead in the wrong direction once a message already is in the proximity of the destination. Therefore, we introduce the directed betweenness measure as a slight modification to the general ego flow betweenness: Instead of all possible paths in the neighborhood of u, only those containing d are considered:

$$\delta_u(d) := \sum_{\substack{v,w \in N_1(u) \\ \mathbf{d} \in N_1(v) \vee v = d}} \frac{(\omega_{u,v} \cdot \omega_{u,w})^2}{\omega_{u,v} + \sum\limits_{\substack{x \in \{u\} \cup N_1(u) \\ x \neq v \neq w}} \omega_{x,v} \cdot \omega_{x,w}} \tag{5.10}$$

5.2.6 Defining an Utility

Keeping in mind the considerations of the previous section, we define an utility for the node v with a message bundle for d currently in contact with node n. Depending on the available neighbor information, we weigh the similarity $\sigma_v(d)$ strongest, the destination betweenness $\delta_v(d)$ second and the general betweenness ϑ_v weakest. These constants were determined empirically. A general method to derive these constants is still under research.

$$A = \begin{cases} 0.9 & \text{if} \begin{cases} v \to d \text{ or} \\ v \to n \to d \end{cases} \\ 0.4 & \text{if} \begin{cases} v \to \not{n} \to d \text{ or} \\ n \to \not{v} \to d \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$$B = \begin{cases} 1 - A & \text{if} \begin{cases} d \notin N_2(v) \text{ and} \\ d \notin N_2(n) \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

$$C = 1 - A - B \tag{5.13}$$

$$U_v(d) = A\sigma_v(d) + B\delta_v(d) + C\vartheta_v \tag{5.14}$$

5.2.7 Update Messages

In our implementation for the related work and our own proposed algorithm, routing decisions are made when a new encounter occurs. This may result in situations where a node will not forward a newly received bundle to a still connected and possibly better suited neighbor because of the absence of an encounter notification. In general, this problem should be solved by an underlying routing layer that transparently communicates in the temporarily connected subnetwork using a MANET algorithm. On the other hand, such layer is only useful if there is a meaningful number of connected subnetworks at any time.

We address this issue by introducing an ***Up**date message* scheme. In this scheme, we look up the neighbor table of our connected node. If the node is connected to the destination, the message can be delivered directly to it.

This way of updating does not influence the routing decisions or performance itself, but can be interpreted as a evidence on how well a routing performs on forwarding a bundle to the one hop proximity of the destination, and shows in which situations a subnetwork forwarding strategy might be useful.

Support on the underlying discovery layer can base on an update counter in the discovery beacon, as described in Section 4.5.3. A change to the internal bundle database triggers a new value of the update counter, which in turn triggers a new exchange between neighboring nodes.

5.3 Evaluation

As DTN routing algorithms are still a relatively young research area, best practices in evaluation are still emerging. In the following, we give a short overview over the metrics and trace data sources we used for evaluating SimBetAge.

5.3.1 Metrics

As DTN algorithms are used for communication, the arrival of data is a focal point. The *delivery rate* signifies how many messages out of a theoretical maximum arrive. The received messages may have taken widely different paths. The *path length* and the *delivery time* can be used to characterize these paths. Ideally, both should be minimized, but as we described in Section 2.3.3, a path with more hops might result in faster delivery.

Additionally, depending on the algorithms used, different amounts of meta-data are necessary. This *control message overhead* expresses at which additional communication cost a certain performance can be achieved. This overhead might make the transfer of data messages prohibitively long or energy consuming, resulting in the need of careful evaluation in a given scenario.

5.3.2 Traces

Connectivity over time in delay tolerant networks depend heavily on the use case. Although simulation studies typically make use of random waypoint models (see [KCC05] for an overview over methods in MANET simulation studies), we decided to use traces produced by actual experiments on social structures to properly capture the social behavior of participants. In the following, we elaborate on the traces used and on some of their characteristics.

In the context of the *MIT Reality mining* project [Nat05a], researchers followed a group of 100 subjects at MIT over the course of the 2004/5 academic year, see also Section 2.4.1. It is one of the most comprehensive records of social interactions available at present and was also used by the authors of the original SimBet paper [DH07].

In our experiments, we decided to use only the second half of these traces, as the first half includes only about 4% of the actual contact information. In the following, we call this trace *MIT50*.

Another project collecting traces of human interaction is the *Haggle* project [Jam06], cf. Section 2.4.2. A variety of users in office and conference environments collected Bluetooth sightings. We filtered these traces to only include sightings of subjects actively interacting with others, while filtering out passive devices like Bluetooth headsets, laptops, and so on. In this evaluation, we call this trace *imote*.

Third, the traces generated in the *Dartmouth Outdoor experiment* [GKN+] closely resemble a random waypoint simulation, see Section 2.4.3. A group of people were walking around an athletic field and collecting GPS information. In the beginning and in the end, all the subjects were meeting up at a central location, so that we actually used only the second and third quarter of this trace. In the following, we call this trace *darmouth*.

5.3.3 Results

In this section, we compare seven different routing strategies. The first strategy is *Direct Delivery*. Only messages where the sender itself meets the recipient are delivered. This strategy is used as the base line as the minimum performance a strategy should achieve.

Epidemic Forwarding represents a second strategy, where every message is forwarded to anyone a node meets. This results in a significant message overhead. However, it serves as a theoretical upper bound, in terms of delivery rate.

Next, we consider the original *PRoPHET* algorithm, as described in [LDS04] and the original *SimBet* algorithm as described in [DH07].

Finally, we evaluate different variants or our proposed algorithm. The first variant *SimBetAge* replaces the binary representation of similarity and betweenness, but does not include the directed betweenness, introduced in Section 5.2.5. The next variant, *Dest2SimBetAge*, follows the design exactly as presented in the SimBetAge section, including directed betweenness and strictly local knowledge. This is the variant we propose as the best trade-off. As a final variant, *DestSimBetAge*, we extend Dest2SimBetAge to consider all known transitive links for the directed betweenness calculation, similar to PRoPHET. This variant performs only slightly better than Dest2SimBetAge, but needs more state information.

Figure 5.4 compares the packet delivery rate achieved by the different routing strategies. While our performance is better in all traces considered, the most drastic improvement can be seen in the MIT50 trace, which is incidentally the longest running trace in which the most change happens. In this trace, the delivery rate is 95% better than SimBet and about 6% better than PRoPHET.

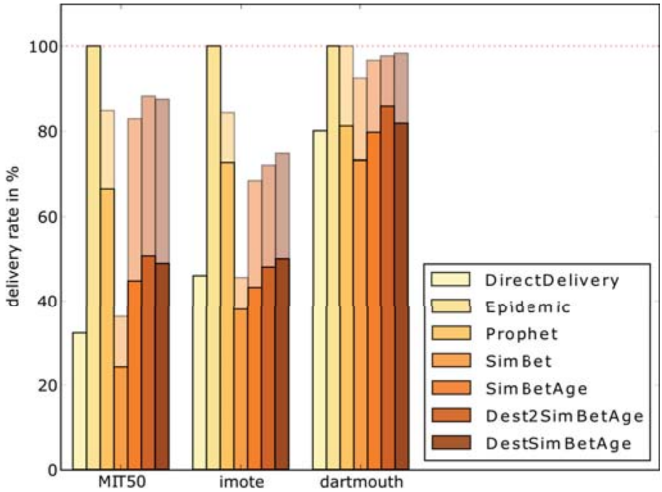


Figure 5.4 Bundle delivery rate for different routings. The paler bars signifies the additional effect of local update messages. We see that SimBetAge, in particular in its Dest2SimBetAge variant, provides a good delivery performance, while only requiring local knowledge.

As we see in Figure 5.5, the hop count distribution of our approach is significantly better compared to PRoPHET, while the delivery rate as shown in Figure 5.6 stays comparable.

With respect to control message overhead and total message overhead (not shown) PRoPHET and our approach are comparable, while we reduce the needed state information, and therefore bytes transferred, significantly, using only an ego-centric representation.

In summary, respecting the age of node relations leads to a better performance than a binary representation. The directed betweenness is a very useful addition to SimBetAge and performs better on all traces. However, the additional global distance information introduced in DestSimBetAge does not have a strong positive effect on performance, thereby supporting our decision on keeping the ego-centric view instead of producing more overhead on discovering nodes not covered by the two-hop neighborhood.

5.4 Conclusions and Outlook

We presented our extensions to SimBet which also take changes in the social structure over time into account. For that purpose, we defined similarity as well as ego flow betweenness in weighted graphs. Furthermore, we evaluated these extensions for the purpose of forwarding messages in delay and disruption tolerant networks, specifically those based on social networks. We showed that the performance can be drastically improved when temporal changes are considered.

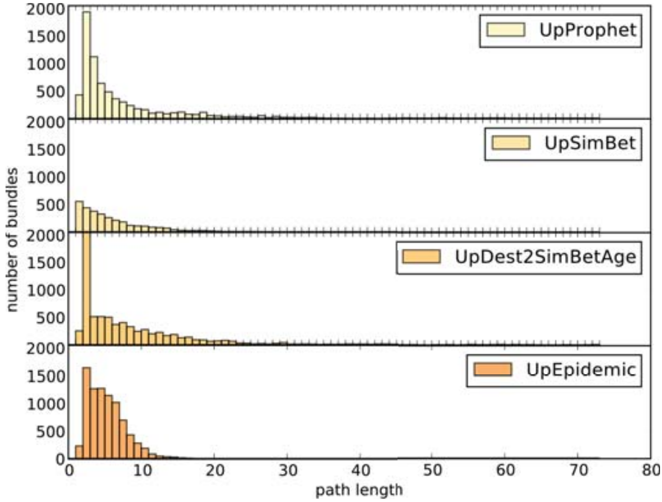


Figure 5.5 Bundle delivery path length for the MIT50 trace.

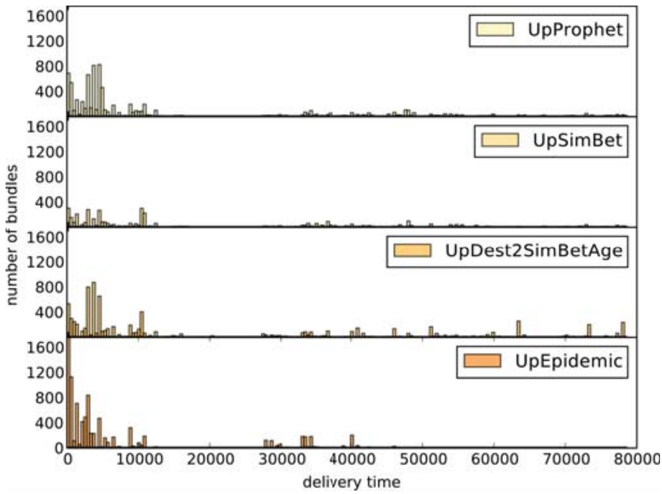


Figure 5.6 Bundle delivery time for the MIT50 trace.

An interesting point we still left open here is the optimal choice of decay and growth parameters as well as how to best mitigate discretization artifacts in the context of implementations in embedded systems.

5.5 Acknowledgments

The problem of social network based routing for DTNs arose out of the RatPack project funded by Microsoft Research. The initial idea of using Daly and Haar's work [DH07] as a potential basis arose in discussions with Elias Weingärtner. The algorithmic design and prototype was built in collaboration with Nicolai Lune Viol, first in his function as a student research assistant and later over the course of his Diploma thesis [Vio09]. We published a preliminary write up of this work at MobiQuitous 2009 [BLVGW09b] and an extended version at the U-NET workshop at CoNEXT 2009 [BLVGW09a].

6

Geography Based Forwarding

In the absence of network infrastructure, DTNs or pocket switched networks [HCG+06] provide communication capabilities to mobile devices using the nodes provided by other network participants to store, carry, and forward messages. While the lack of continuous end-to-end connectivity, and the dependence on typically resource constrained devices present us with a plethora of challenges, this fundamentally different type of communication network offers a chance for next generation networks to provide data connectivity to areas, where infrastructure is either too expensive or otherwise infeasible.

Previous approaches mainly use social characteristics [DH07], statistics of meetings between nodes [LDS04] or other heuristics [SPR05], we base this scheme on the notion of location distribution. Upon the meeting of two nodes, our approach compares their location distributions and chooses the subsequent carrier for a message bundle accordingly. Using this approach, we are able to make meaningful forwarding decisions with only local knowledge even if an intermediary node has no knowledge of even the existence of the recipient.

Contributions

The main contributions of this chapter are:

1. **Modeling human mobility:** We model human mobility as a probability distribution around known anchor points. With this, probabilities for future locations can be quantified.

2. **Routing heuristic based on location distribution:** Based on this model, we predict future connectivity and present a heuristic for disruption tolerant networking, specifically addressing regional and global mobility.

3. **Trace of self-reported global mobility:** The advent of location based online social networks allowed us to collect global location traces for different classes of users.

6.1 Related Work

A multitude of disruption and delay tolerant routing schemes have been proposed, see also the previous chapter. In general, proposed routing strategies can be categorized in four main categories:

- **Statistics:** Only statistics of past node encounters are used to build a model of future connectivity.
- **Social network:** In addition, these strategies make assumptions based on human behavior.
- **Geography based:** These strategies make use of the geographic location information of nodes in a network.
- **Other heuristics:** Other strategies exist, e.g., when future connectivity is well known, e.g., in space based communication.

A prominent instance of the statistics based approach is PRoPHET [LDS04], which predicts future delivery by modeling links between nodes as a probability, rising on each meeting, but aging during inactivity. The viability of a path is then measured as the product of all intermediate links. This approach assumes information on every link being available to all nodes, thus limiting its scalability to well-defined regions.

Social network based approaches, like SimBet [DH07] are based on social network characteristics like the similarity of a node with the destination of a message bundle and the betweenness of a node compared to its neighbors. SimBetAge, see Chapter 5, adds the concept of aging as in PRoPHET. While these approaches depend only on local knowledge, their scalability is still limited by the rise of local maxima in social networks over a certain size.

This work is not the first to explore the practicality of geographic information for disruption tolerant routing. *GPSR* greedily forwards messages closer to the destination in an urban city environment [KK00]. However, it assumes storage nodes at every (street) intersection on the path, making it infeasible in practice. *Move* [LCGZ05] builds on the heuristic that if a node is moving towards the destination, it is better suited to forward messages. In *MobyPoints* [LFC05] a vector encodes meeting probabilities for all possible locations. They then use the similarity score between these vectors for forwarding decisions. Nagy et al. [NS07] provide a more general survey of location based approaches.

Routing mechanisms tend to optimize message transfer per message. However, as Pujol et al point out in [PTR09], this typically puts a significant load on very few central nodes, draining their energy unacceptably fast. Fairness in load distribution is therefore another important aspect to consider.

The core idea of our approach bases on the predictability of human mobility, which Song et al. [SQBB10] impressively showed using data which was collected tracking the mobile customers of a cellular GSM network. Previously, Brockmann et al. showed that human travel roughly follows a power law distribution [BHG06], a notion we also consider in our approach.

6.2 Design of GeoDTN

GeoDTN bases its routing decision on previous node movements with a probabilistic node meeting heuristic. The nodes' movement patterns are reactively compared to the destination's pattern resulting in a transitive probabilistic meeting score $s_{i,dest} \in [0,1]$, denoting of how probable it is that node i and the destination node have a common movement domain.

6.2.1 Mobility Representation

The node mobility representation extends the approach of MobyPoints [LFC05] that each node has a mobility vector containing all necessary information for the routing decision. The locations of observed nodes are collected into *clusters*, represented by a bivariate normal distribution and a confidence. For n clusters, each with j_m locations, the probability for cluster m is:

$$p_m = \frac{j_m}{\sum_{i=1}^{n}(j_i)}$$

We model the confidence $\zeta \in [0,1]$ of a cluster as an exponential decay function:

$$\zeta(\Delta t) = e^{-\alpha \cdot \Delta t}$$

with Δt denoting the time since the last update and α an aging parameter. Every time the node visits a location in the corresponding cluster, we reset ζ to 1. This corresponds to $\Delta t = 0$. This approach is similar to the decay function we use in our SimBetAge work. However, in GeoDTN as opposed to SimBetAge, we separate the confidence ζ from the probability p.

All nodes exchange the mobility vectors of their 2-hop neighborhood upon contact. Therefore a node n_i may receive a mobility vector of another node n_j several times. These vectors may have different confidence values for the same clusters or include new clusters. Therefore the mobility vector of a node n_i may not be consistent between neighboring nodes.

If node n_i receives a mobility vector v from node n_j, we consider the following cases:

1. **$\mathbf{n_i}$ receives the mobility vector of node $\mathbf{n_j}$, $\mathbf{v_{n_j}}$.**

   As node n_j is the authoritative source for information on itself, node n_i updates all its information accordingly.

2. **$\mathbf{n_i}$ receives $\mathbf{v_{n_k}}$ which it already has.**

   Node n_i compares all clusters in the mobility vector v_{n_k} with the ones received from n_j keeping the cluster with the higher confidence value ζ.

3. **$\mathbf{n_i}$ receives $\mathbf{v_{n_k}}$ which it does not have yet.**

   n_i has no previous knowledge of v_{n_k} and therefore adopts the complete vector v_{n_k} from n_j.

Node n_i prunes the clusters in the mobility vectors of neighboring nodes when their confidence drops under a fixed threshold ϵ_ζ. Entries in its own mobility vector are pruned using an adaptive threshold ϵ_p. Node n_i calculates this threshold using the mean over all probabilities in its mobility vector $\bar{p}$ and its standard deviation σ_p^2 using the following formula, as well as a spreading factor γ:

$$\epsilon_p = \bar{p} - \gamma \cdot \sigma_p^2$$

6.2.2 Neighbor Score

We define two nodes as being neighbors if they are in radio range at a common location frequently. To determine this, we define a neighbor score using the mobility vector v_{n_i} of a node, with each entry consisting of the bivariate distribution represented by $f_{i,k}(x,y)$, parametrized by $\bar{x}$, $\bar{y}$, σ_x^2, σ_y^2 and ρ, as well as its probability $p_{i,k}$ and its confidence $\zeta_{i,k}$.

The movement score of an element $e_{i,k}$ of the mobility vector of node n_i within the area $([x_{\min}, x_{\max}], [y_{\min}, y_{\max}])$ can thus be represented as:

$$s_{e_{i,k}}^{\text{local}} = \int\limits_{x_{\min}}^{x_{\max}} \int\limits_{y_{\min}}^{y_{\max}} f_{i,k}(x,y)\, dx\, dy$$

Using this, two nodes n_i and n_j can calculate their meeting score upon meeting. For this, they calculate their common movement area as the intersection of their own movements. Integrating only over the common part of the distribution, and not the complete distribution captures the strength of the overlap. The neighbor score can thereby be defined as:

$$\text{Neighbor Score} = \frac{\text{Movement in Common Movement Area in } [x_{\min}, x_{\max}] \text{ and } [y_{\min}, y_{\max}]}{\text{Complete Movement Area of Both Nodes}}$$

The actual calculation also weights the probabilities p of the actual entries in the mobility vector as well as its confidences ζ.

$$s_{i,j} = \frac{\sum\limits_{e_{i,k} \in v_i} \zeta_{e_{i,k}} \cdot p_{e_{i,k}} \cdot s_{e_{i,k}}^{\text{local}} + \sum\limits_{e_{j,l} \in v_j} \zeta_{e_{j,l}} \cdot p_{e_{j,l}} \cdot s_{e_{j,l}}^{\text{local}}}{|v_i| + |v_j|}$$

Two nodes are neighbors if their neighbor score $s_{i,j}$ within their common area reaches a neighbor threshold ϵ_n.

6.2.3 Routing

The routing decision of geoDTN bases on the neighbor scores $s_{i,\text{dest}}$. Similar to SimBetAge in the previous chapter, we distinguish three different cases, depending on how far a node is from the destination. When node n_i comes into radio range of node n_j the neighbor scores $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$ are compared. If both are below a threshold, we employ (1) a *distance mode*. If a message is stuck in a local minimum, we employ (2) a *rescue mode*. Finally, when a node is in the vicinity of the destination, we use (3) *scoring mode*.

6.2.3.1 Distance mode

If both $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$ fall below the scoring mode threshold, a bundle is forwarded into the neighborhood of the message destination following a hill climbing strategy. The distance of node n_i to the destination n_{dest} is the minimum distance between their mobility vectors weighted with confidence ζ and probability p:

$$d_{i,\text{dest}} = \min_{\substack{f_{i,k} \in v_i \\ f_{j,\text{dest}} \in v_{\text{dest}}}} (m \cdot \text{dist}(f_{i,k}, f_{j,\text{dest}}))$$

The factor m increases the distance for elements of the mobility vector that have only a small probability or a low confidence and is defined as:

$$m = 10 \cdot e^{-\beta \cdot \zeta \cdot p}$$

It favors elements with a high probability p and a high confidence ζ. The factors 10 and β are used to adjust this weighting.

6.2.3.2 Rescue mode

As distance mode employs a hill climbing strategy, message bundles may get stuck in a local minimum. To counteract this problem, we employ random walks. Thus, if a node n_i detects a stuck bundle, this bundle b is then forwarded randomly within the network until the distance $d_{b,\text{dest}}$ of this bundle is either smaller than $d_{i,\text{dest}}$, the bundle reaches scoring mode distance to n_{dest}, or a maximum number of hops ϵ_h is reached.

We consider two variants for random forwarding of a bundle:

1. Upon contact with another node the bundle is forwarded with a probability of $p = 0.5$, or

2. If this bundle already passed through a particular node, we forward that bundle back to it with probability $p = 0.3$. If this bundle has not passed through the particular node, we forward it to that node with a probability $p = 0.7$.

The second variant allows the message bundle to explore new network parts with a higher probability and thus gets out of the local minimum network faster.

Detection of stuck bundles bases on a time threshold ϵ_s and a log of all nodes previously visited by that bundle. Since a node is aware of its neighborhood, it can detect if it tried all neighbors and if ϵ_s elapsed since it received the bundle. A bundle is therefore considered stuck, if within the time threshold ϵ_s, a bundle arrives a second time at a node.

p_m	probability of a location cluster m
ζ	confidence in the information about a cluster
Δt	the time which passed
n_i	a node
e	a mobility vector element, parametrized by $\bar{x}$, $\bar{y}$, σ_x^2, σ_y^2 and ρ, p, and ζ
f	bivariate distribution defined by a mobility vector element
v	a set of mobility vector entries f
s	movement score between two mobility vector elements
d	distance between nodes
ϵ_n	neighbor threshold
ϵ_ζ	confidence threshold for pruning other information on others
ϵ_p	adaptive threshold for pruning own information
ϵ_h	maximum hop count after which messages enter rescue mode
ϵ_s	stuck detection threshold
m	distance correction factor
β	exponential correction factor
n_{dest}	scoring mode distance

Table 6.1 List of symbols and abbreviations used in this chapter

6.2.3.3 Scoring mode

Finally, when a bundle reaches the vicinity of the destination n_{dest}, we use a scoring mode to select the best possible next hop. When node n_i carrying the bundle comes into range of node n_j, they both calculate their respective neighbor scores with the destination $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$. Additionally, they calculate the neighbor scores of their neighborhood, weighting them with their own score. Node n_i therefore forwards the bundle to n_j, if:

$$\max\left(\{s_{i,\text{dest}}\} \cup \{s_{k,\text{dest}} : n_k \in N_i(2)\}\right) < \max\left(\{s_{j,\text{dest}}\} \cup \{s_{l,\text{dest}} : n_l \in N_j(2)\}\right)$$

Using this scheme, the bundle is continuously forwarded closer to the destination, using the best possible path based on the mobility patterns of the encountered nodes.

On a final note about addressing, we intentionally leave out this problem in this work, assuming the sending node is aware of the destination's mobility vector. We also anticipate that an iterative refinement of the destination's mobility vector, while the bundle traverses the network and nodes are more aware of the destination's actual mobility, leads to good results.

6.3 Evaluation

We evaluate geoDTN based on two different scenarios, both introduced in Section 2.4. First, we use *UMass DieselNet Fall 2007* traces [BLM+08] containing GPS movements of 33 nodes for 18 days, where we assume a simulated radio range of 150 meters. This range was chosen as a typical outdoor WLAN range.

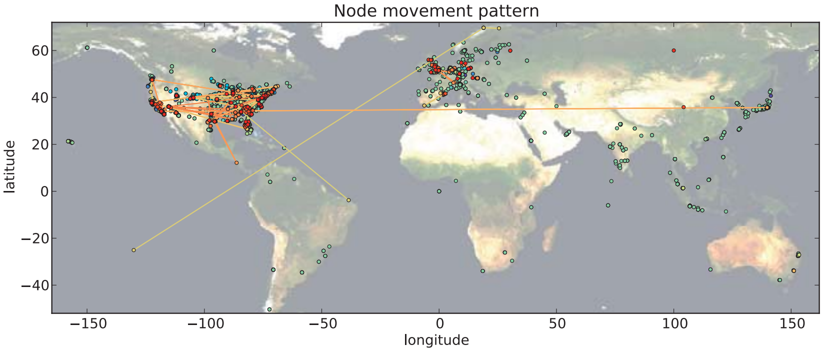


Figure 6.1 Resulting network from the observation of 221 BrightKite users over the course of two months. For readability, we also display connecting lines for a sample of 50 users.

Secondly, we use a human based network trace making use of the social networking site *BrightKite*. We follow the *check-ins* of 221 very active users over a period of two months assuming a contact, whenever they come within 300 meters of each other. Hence, these contacts are based on real human behavior, which for the purpose of this paper are assumed to be 221 test users making use of pocket switched networking devices. The resulting network is shown in Figure 6.1. As BrightKite is a photo sharing community and doesn't track users aside from picture meta data, we somewhat arbitrarily chose 300 meters to capture the movement before and after a picture was taken as a proxy for unavailable more exact movement information.

From the real world timing, we generate discrete events such that between node actions, e.g., meetings and position changes, there are between 5 and 15 events. This filtering allows us to run a simulation study in reasonable time. However, filtering neither changes location order nor prominence. After filtering, we obtain 43786 discrete time events.

6.3.1 Traffic Model

The assumed traffic model generates a message from each node to d random destinations every t time units. For evaluation purposes, we assume timing intervals $t \in \{5, 10\}$ and the number of destinations $d \in \{2, 3, 5, 7, 10\}$. Those numbers mirror assumed probabilistic usage patterns.

6.3.2 Simulation and Metrics

The simulation uses static message and node trace files within a custom python simulation software. Assuming a trace file length of 5000 time events, we allow a learning period of 2000 time events, upon which message generation takes place for 500 time events. Finally we allow messages another 2500 time events to arrive at their destination. Using static message and node trace files enable us to repeat experiments accurately.

Figure 6.2 Evaluation of the delivery time, delivery rate, and hop count in UMass DieselNet.

Our key metrics are delivery rate, delivery time and hop count. In addition to that we also consider the energy consumption per node and received bundle, as well as the fairness, i.e. how fast more central nodes are drained of energy.

Epidemic routing with unlimited queue length and transmission capacity provides us with the theoretical upper bound in terms of delivery rate, and the lower bounds in terms of delivery time and hop count. In contrast, Direct Delivery—only forwarding messages to the destination when the destination is directly in radio range—provides us with a sensible upper bound for the delivery time, and a sensible lower bound for the delivery rate as any slower or less effective algorithm is evidently flawed. At the same time, it is a true lower bound in terms of hop count—precisely 1.

Additionally, we compare geoDTN with PRoPHET, SimBet, SimBetAge of the previous chapter, and Move. Furthermore, we evaluate geoDTN with (*geoDTNrand*) and without (*geoDTN*) rescue mode.

A detailed parameter evaluation using hill climbing strategies to optimize parameters of the DieselNet trace files results in the following optimal parameter settings: Aging parameter $\alpha = 0.7$, neighborhood score $\epsilon_n = 0.0009$, and distance threshold $\epsilon_d = 0.0005$.

6.3.3 Performance in DieselNet

Referring to Figure 6.2, we observe a delivery rate of approximately 85% for both variants of geoDTN, very similar to the results of SimBet and SimBetAge, still

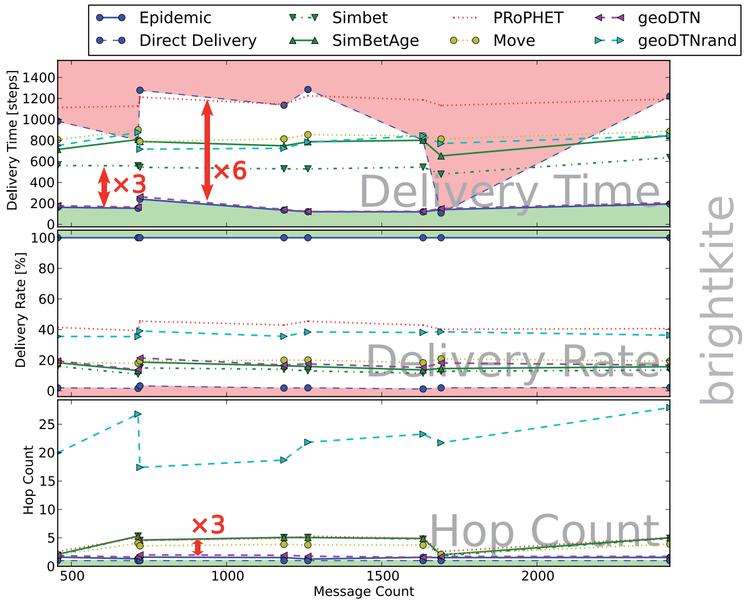


Figure 6.3 Evaluation of the delivery time, delivery rate, and hop count in BrightKite.

outperforming the simpler Move. PRoPHET however consistently outperforms our approach by a margin of 5% in terms of delivery rate, underlining the quality of PRoPHET as one of the best state of the art DTN routing algorithms with respect to delivery rate.

Regarding hop count, geoDTNrand needs an average of 1.2 more hops than geoDTN to the benefit of a 2% higher delivery rate. PRoPPHET with an average of 2.2 more hops and Move with 5.1 more hops are significantly outperformed by our approach, whereas SimBet and SimBetAge are nearly optimal with respect to this metric, with SimBetAge performing best—aside from the theoretical boundaries.

The delivery time of geoDTN is worse than the other algorithms we considered, while SimBet and SimBetAge outperform them. This indicates that our geoDTN variants occasionally miss the optimal point in time for forwarding a message. Deeper inspection of the log files reveals the reason for this slightly lesser performance: The transitive score calculation leaves room for further improvement.

6.3.4 Performance in BrightKite

Figure 6.3 shows the evaluation results for the BrightKite traces. The simulations use the same parameter as before.

Like in the DieselNet traces, geoDTN's performance in delivery rate is similar to SimBet and SimBetAge with geoDTN being slightly better. GeoDTNrand performs significantly better than geoDTN at the cost of a very high hop count. Nevertheless,

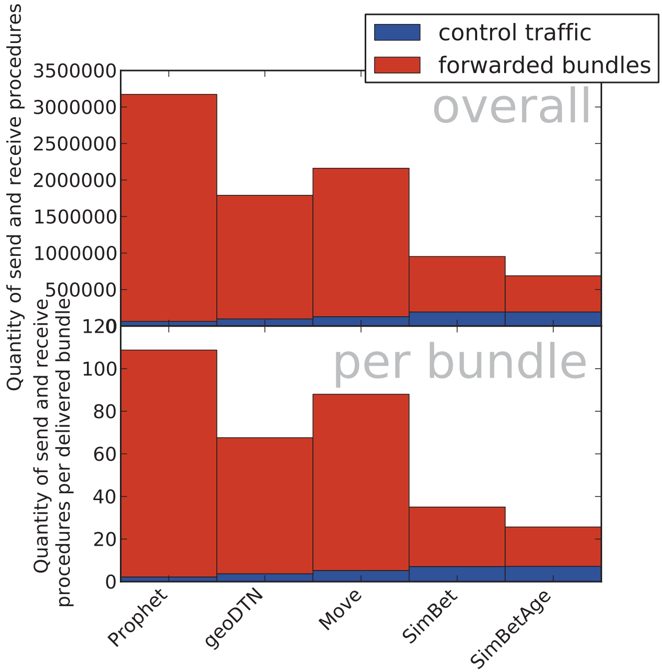


Figure 6.4 Quantity of sent and received messages overall (top) and per received bundle (bottom) in DieselNet. Control messages (blue) and message bundles (red) are counted separately.

the rescue mode shows itself to be a reasonable extension to improve delivery metrics. The hop count can be further reduced when optimizing the random walk parameter for BrightKite like scenarios.

PRoPHET outperforms the other algorithms but delivers only about 2% more packets than geoDTNrand. Move performs better in this scenario in contrast to the other algorithms. This indicates that taking the angle of movement into account is a reasonable indicator for routing decisions in networks based on human mobility.

In this scenario, geoDTN (not geoDTNrand) significantly outperforms the other algorithms in terms of hop count and delivery time. On average, it performs 230% better in terms of hop count, with individual cases reaching 300%. With respect to delivery time, it outperforms SimBet by a factor of 3, SimBetAge and Move by a factor of 4 and PRoPHET by a factor of almost 6.

Therefore, we show that geoDTN performs well in this scenario based on human mobility on a global level scale.

6.3.5 Overall Energy Consumption

Figure 6.4 shows the energy consumed per simulation run for each routing algorithm. We distinguish between the overall energy consumption and the per received bundle energy consumption. Each *send and receive* action raises the used amount of energy by one, assuming constant message size.

Since we model message forwarding as taking place in a broadcast medium, energy consumption counts towards all nodes in radio range, not just the node a message was addressed to on this hop.

As a growing hop count results in additional send and receive actions, PRoPHET and Move display the highest energy consumption. Counterintuitively, Move consumes less energy than PRoPHET although it has a higher average hop count. The reason for that lies in that PRoPHET uses nodes in more central and dense areas in the network to a greater extent. This results in more nodes receiving bundles not addressed to them and therefore a higher energy consumption.

The amount of control messages is inversely related to the hop count. This supports the intuitive assumption that more numerous and more recent control data is useful to find optimal paths in dynamic networks.

6.3.6 Fairness

The fairness evaluation is a measure of the energy distribution over the different nodes and is related to the fairness approach of Pujol et al.. We define fairness such that all nodes need approximately the same amount of energy instead of exploiting a small subset of—more central—nodes.

Figure 6.5 shows the energy needed per node in the DieselNet evaluation. SimBetAge followed by SimBet clearly outperform the other algorithms with low overall energy consumption and a relatively fair distribution over the different nodes.

However, geoDTN outperforms Move and PRoPHET in this scenario as well. PRoPHET, in comparison to all other algorithms we investigated, fails being energy efficient as well as being fair. It thus depletes the most central nodes the fastest lowering the incentive for central users to participate.

6.4 Conclusion

We presented a novel disruption tolerant routing scheme based on the mobility of the participating nodes. The scheme employs three routing modes. *Distance mode* routes a packet by minimizing the distance to the destination on the path to the destination node's neighborhood. In case of failure, *rescue mode* assures that a message bundle makes progress when stuck in a local minimum. Within the destination's neighborhood a probabilistic *scoring* function determines the similarity of a node's movement pattern with that of the destination's with respect to probability and confidence in the stored information. The *neighbor score* $s_{i,j} \in [0,1]$ allows

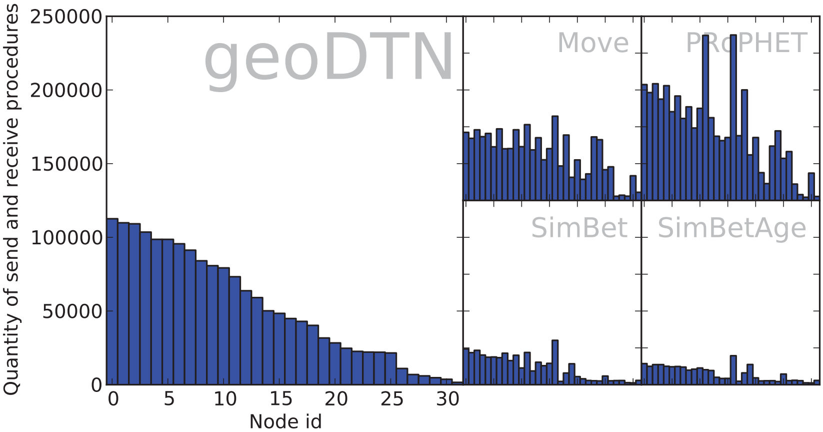


Figure 6.5 Fairness evaluation on the UMass DieselNet traces on an equal scale, with a zoom into geoDTN for focus.

the comparison of mobility vectors of different nodes, taking the *probability* and *confidence* of individual locations into account.

The evaluation shows that geoDTN works similarly well as algorithms based on social group characteristics such as SimBetAge and outperforms binary movement algorithms as Move. In dynamic, less periodic networks, geoDTN significantly outperformed the other algorithms in hop count and delivery time. It performs in general 130%, in individual cases even 200%, better in hop count. In terms of delivery time, it outperforms the other algorithms on average by a factor of 3, PRoPHET by factor 6. This demonstrates geoDTN to be a fast self-adapting algorithm for human mobility based networks.

The energy and fairness evaluation shows that geoDTN performs significantly better than the best state of the art algorithm in delivery rate, PRoPHET. GeoDTN needs less hops, thus less transmission energy and distributes the load fairer over the different nodes.

6.5 Acknowledgments

The problem of geography network based routing for DTNs arose out of the RatPack project funded by Microsoft Research as an extension and follow up of SimBetAge in the previous chapter. The algorithmic design and prototype was built in collaboration with Daniel Schmitz over the course of his Diploma thesis [Sch10a]. We published this work at GLOBECOM in 2011 at the Next Generation Networking Symposium [BLSW11a].

7

DTN Applications

Platform vendors such as Google, Apple and Microsoft increasingly recognize the importance of offline operation. While Apple only asks of iOS developers to handle network failures gracefully [App16], Microsoft gives more detailed instructions for its Windows Apps [Win16]. In contrast, Google includes offline capabilities as part of its *Building for Billions* campaign [And16] for Android, that aims at making mobile applications useful also with “low, intermittent, or expensive connectivity”.

With local low power peer to peer type communication technologies such as *Bluetooth Low Energy* finding their way into off the shelf consumer electronics, the technical barriers for opportunistic networking disappear. While the use of perfect difference sets for neighbor discovery, see Chapter 4, can further improve energy usage during disconnected periods of time, the introduction of high throughput local communication technology such as *Wi-Fi Direct* enables the opportunistic transfer of larger data items, even during short connections. Approaches such as social network or geography based networking, see Chapter 5 and Chapter 6 respectively, address the forwarding of messages across disconnected areas.

Having the fundamental building blocks of disruption tolerant networking in place and being aware of the influences on a user’s quality of experience, see Chapter 3, we now explore services and applications that lend themselves to this type of networking. While we continue to argue that there is no singular killer app for DTNs but rather that the majority of network services and applications benefit from becoming more aware of the network conditions and general user context and being able to forward data opportunistically, we discuss a selection of example use cases and a potential architecture for disruption tolerant networking.

7.1 Previous File Sharing Approaches

Previous opportunistic networking applications targeted at consumers often focus on file sharing. As an example, *BlueTorrent* [JLCCG07] is a cooperative P2P file sharing application based on Bluetooth. Discovering nodes in the vicinity using Bluetooth scanning, it then exchanges files in that the peers are both interested. The authors of BlueTorrent try to find optimal settings for the initial Bluetooth inquiry and the connection establishment, but the approach still suffers from the energy hunger of the Bluetooth device discovery.

A similar approach is [HYKPK10] based on Ad-hoc-networking for neighbor discovery and data exchange. While Ad-hoc networking is not generally supported in current mobile devices, the approach of advertising interest locally and accepting incoming files that match this interest is very well suited for opportunistic networking.

Finally, *Penumbra* [Gre07] also forwards files opportunistically with Wi-Fi devices on the same channel making use of extensions to IEEE 802.11 beacon frames to announce interests, file availability and file contents. The author designed Penumbra to work anonymously in the background while devices are at the same time connected to an infrastructure network. Energy consumption was not considered, as the devices are assumed to be connected to a power source in an office or private household environment.

In the following, we will argue for a more generalized architecture and a set of sample applications that make use of this architecture.

7.2 DTN Service Architecture

The most well-known DTN service architecture right now are the DTN2 reference implementation [DD10], ION [Bur07], IBR-DTN [SMPW11]. While they all aim at implementing the bundle protocol [RFC5050] as interoperable as possible, their goals and trade-offs slightly differ.

The DTN2 reference implementation[1] is the primary implementation for the bundle protocol, on which the majority of protocol extensions and prototypes were based. It is very flexible with many configuration and debug options. The flexibility, however, comes at the prize of many software dependencies and a relatively steep learning curve. DTN applications use a Sun RPC to interact with the main daemon.

IBR-DTN primary focus is on portability and resource constrained systems while keeping external dependencies at a minimum. Running on stable on highly constrained OpenWRT and embedded Linux platforms, it slowly becomes the implementation of choice in the community for researchers running active deployments instead of experimenting on the protocol design itself. DTN applications interact with IBR-DTN either via a socket based on the TCP Convergence Layer, or a specialized API, meaning that an application developer needs to build fully compliant bundle protocol messages.

ION (Interplanetary Overlay Network) was written by the Jet Propulsion Lab specifically for use in communication with spacecrafts. As such, its requirements are more targeted, while the main focus is reliability and not the ease of implementation of third party DTN application developers. It runs on VxWorks and Linux platforms.

[1] `https://sourceforge.net/projects/dtn/`

Tag	Semantic	Example
replyTo	Specifies to where to send reply packets	replyTo_userA
deliverTo	Specifies intended recipient	deliverTo_userB
hasPlugin	Indicates that a given plugin is required for the packet	hasPlugin_wikipedia
tr_sticky	Specifies to keep a local copy even after a successful transmission	tr_sticky
* (Wildcard)	Matches any tag if compared against	deliverTo_*

Table 7.1 A selection of tags interpreted by the ADAMA architecture. Service or transport plugins may specify and interpret arbitrary additional tags.

We argue that while the underlying groundwork and protocol may be useful, to truly enable application developers to improve their services by enabling them in disconnected or opportunistic network environments, we need to improve the API. This leads to a trade-off between two requirements:

- **Simple and easy to understand:** The API should have easy to understand primitives, that can easily be combined while abstracting away common and low-level tasks.
- **Expressive:** the API needs to be expressive enough to implement services without rewriting core code.

The actual GUI for an end user will be highly service-dependent. However, the architecture should neither require the end-user to install complex and maintain software nor force the use a specific operating system.

7.2.1 Design of a DTN API

There are two functions main functions, that the DTN architecture needs to support with respect to applications and services that make use of it: (1) allow the submission of messages into the DTN system with metadata, and (2) inform local service applications of newly available messages. With these two main responsibilities in mind, we designed *ADAMA*, an Architecture for Disruption-Aware Mobile Applications.

As a first consequence, we completely separate the application programming interface from the underlying transport. The underlying transport agent can make use of arbitrary protocol mechanisms. In fact, it could make use of the bundle protocol, or less common transports such as MIME over HTTP [WHFP09], or even uucp [Now78], one of the original transport mechanisms for email before the advent of TCP. The implementation of these transport agents, we call *transport plugins*.

To avoid additional protocol library dependencies for application providers, we made the choice to *separate the payload from the control information* instead of using a specific archive or bundling format. Control information is added in a simple text based sidecar file using the conventions of `.ini` files.

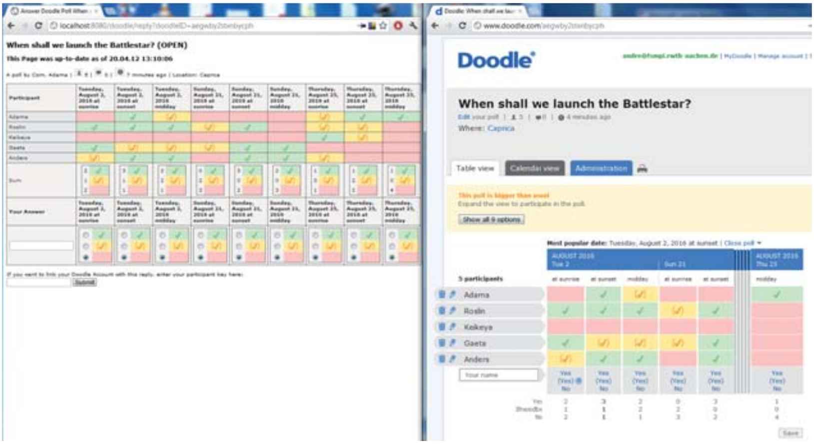


Figure 7.1 Left: A doodle poll rendered by ADAMA. Right: The default view of the same doodle poll via its native web interface.

As soon as an application developer is able to move files into a particular *submission directory* and create a simple sidecar file describing the intended recipient, the developer is able to send messages via a DTN. This control files use *tags* to describe, e.g., the destination, a reply address, or the requirement of a specific ADAMA plugin, see Table 7.1.

Current applications typically do not support DTN concepts directly, but use service specific protocols. An example is email, which can be submitted via SMTP and received using IMAP. While the concept of email is inherently compatible with DTN, neither SMTP nor IMAP are. To allow existing email applications to make use of DTNs, we design ADAMA *service plugins*. This plugin locally opens an SMTP port and an IMAP port. Existing email applications can be configured to use these ports, which in turn handle the translation to the DTN system.

In the following, we describe two service plugins: Doodle and Wikipedia. Both are popular web applications and as such currently require an active relatively low latency web connection. However, by redesigning their application layer protocol and minimal changes to their UI, we are able to use them in disconnected environments.

7.2.2 Doodle

Doodle is a well-known web application that allows groups of users to schedule a meeting or choose between options. The web application makes heavy use of JavaScript and JavaScript Object Notation. However, the necessary data to participate in a poll is very limited. It includes: (1) the name of the poll and potentially place, (2) the name of the organizer and his reply to address, and (3) the available options. To further allow a user an overview of already cast votes, we can further include: (4) the name and choice of each previously known participant, (5) any comments from previously known participant, and (5) the time of the last update.

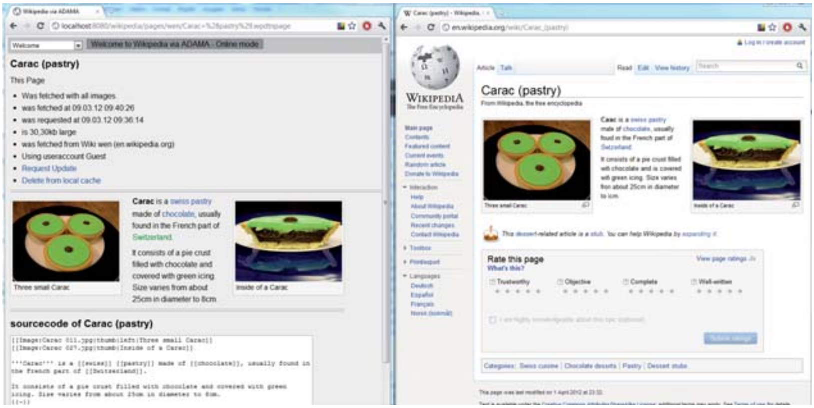


Figure 7.2 Left: A Wikipedia article rendered by ADAMA. Right: The default view of the same Wikipedia article via its native web interface.

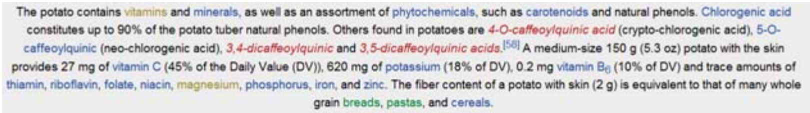
Our service plugin opens a local TCP port running a web server and then notifies users of incoming poll request via a system notification. See Figure 7.1 for an example poll. As discussed in Section 3.2.2, Doodle lends itself well to disconnected operation as it has soft time constraints and lenient order of delivery constraints from the perspective of the application itself.

We did a feasibility test with 12 test users artificially adding delay of up to 500ms and packet loss of up to 10% to their connection. While there was no clear preference for either the native web application nor the ADAMA version, ADAMA did at least not perform worse. This is supported by the following quote of a test user: *"I'd like to see ADAMA in a broader test scenario one day, since I feel that there is lots of potential that ADAMA has not yet been able to show."*. André Goliath reports more details of this feasibility test in his Diploma thesis [Gol12].

7.2.3 Wikipedia

Wikipedia is the single most popular reference and education resource on the web. Many efforts are underway, to make it available in low connectivity environments. One such effort that allows reading and to a certain extent searching is the mapping to an interactive media DVD by Microsoft research in Bangalore [IPT12], that students can browse using low cost DVD players. The main advantage of this approach being that it uses only hardware that is already available also in very low income households with a distribution system already in place—the open markets where DVDs are sold.

In contrast, our approach aims at enabling users to retrieve and edit pages in Wikipedia and other MediaWiki based deployments. Based on the MediaWiki API, we have full access to a wiki's contents, allowing us to read, create, edit and search articles. The main task, from a network point of view is bundling up the complete wiki source code with its embedded images, JavaScript, styles, and media resources. We can inline JavaScript and CSS by replacing the tags that reference external



The potato contains vitamins and minerals, as well as an assortment of phytochemicals, such as carotenoids and natural phenols. Chlorogenic acid constitutes up to 90% of the potato tuber natural phenols. Others found in potatoes are *4-O-caffeoylquinic acid* (crypto-chlorogenic acid), 5-O-caffeoylquinic (neo-chlorogenic acid), *3,4-dicaffeoylquinic* and *3,5-dicaffeoylquinic acids*.[58] A medium-size 150 g (5.3 oz) potato with the skin provides 27 mg of vitamin C (45% of the Daily Value (DV)), 620 mg of potassium (18% of DV), 0.2 mg vitamin B_6 (10% of DV) and trace amounts of thiamin, riboflavin, folate, niacin, magnesium, phosphorus, iron, and zinc. The fiber content of a potato with skin (2 g) is equivalent to that of many whole grain breads, pastas, and cereals.

Figure 7.3 Links to other wiki pages are color-coded by the plugin. Blue pages exist but are not cached yet. Yellow pages are requested but have not arrived yet. Green pages are already available locally. Red pages do not exist yet.

resources with the resources themselves. For JavaScript and CSS, this leads to slight overheads, as different articles can no longer share the same—potentially cached—resource. Images can be embedded in a similar way by replacing them with so-called data URIs [RFC2397] that embed a base64 encoded version of the image in the source code.

Similar to the Doodle plugin, the Wikipedia service plugin opens a local TCP port running a web server. See Figure 7.2 for an example article. Notice that there is an additional meta section at the top as well as the wiki source code at the bottom allowing immediate updates by the user.

This web server in turn rewrites all links to other wiki articles to a corresponding local link. If the user requests a page that has not been cached yet, it will request it from the server. As a further enhancement, all directly linked articles from a displayed article up to a configurable link depth will be fetched concurrently with the initial article request. As a hint for the user, which links are immediately accessible, the plugin recolors links according to their availability, see Figure 7.3.

We performed the same initial feasibility study with the same 12 test users that we used for the Doodle feasibility study with the result being equally vague: While there was no clear preference for either the native web application nor the ADAMA version, ADAMA did at least not perform worse. More details are available in the Diploma thesis of André Goliath [Gol12]. Once the plugins are more mature, a more comprehensive study should be performed.

7.3 FootPath

Opportunistic networking is particularly relevant for mobile nodes. Keeping with the theme of context aware applications, in this section, we investigate user navigation in the physical world, in particular in indoor environments. We describe this approach in more detail in [BLSVW11]. Returning to the enabling of disruption tolerant services, we then investigate potential map dissemination schemes.

While navigation systems for outdoor environments are readily available, indoor navigation is still considered challenging. The main reason for this lies in the difficulty to obtain accurate position information in an easy to set-up way with minimal infrastructure and to create indoor maps.

Our approach to this problem is twofold: (1) We use simple step detection and step heading estimation. (2) We match detected steps onto the expected route from

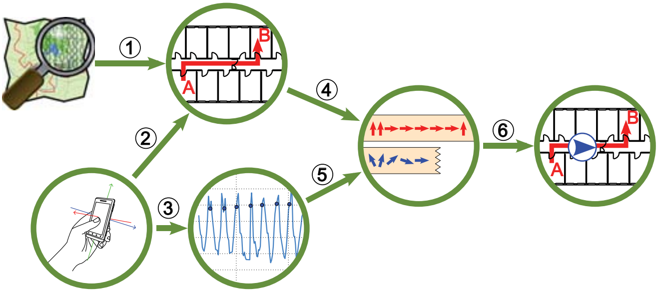


Figure 7.4 Flow of information during navigation. (1) The application obtains map material from OSM. (2) The user selects her current position and her destination. The phone calculates the best route. (3) The mobile phone detects steps and directions. (4) The route is transformed into expected steps. (5) The detected steps are mapped onto the expected steps. (6) The user gets feedback about her position and her next way-points.

the source to the destination using sequence alignment algorithms. Instead of the more general localization problem, we solve the simpler more constrained localization problem on a specified route. This allows us to compensate for inaccuracies and give the user accurate turn-by-turn directions.

To allow easy incremental deployment of our system, we integrate our system with OpenStreetMap [Ope11b], which already has rudimentary indoor support [Ope11a]. GPS, Pseudolites, UWB, WiFi access points, and RFID is avoided, making the system useful for protected environments like historical buildings and archaeological sites as well as hospitals, where additional RF gear might interfere with medical equipment.

Our main contributions are:

1. **Infrastructureless indoor navigation:** We use simple step detection and step heading detection, which we then map onto a route using sequence alignment algorithms. Additional infrastructure, like GPS, Pseudolites, UWB, Wi-Fi access points, and RFID can be avoided.
2. **Localization on a route:** We know the route, the user intends to take. Using this knowledge, we reduce inaccuracy at corners opposed to further accumulating errors. Path matching is precise enough to allow for accurate indoor turn-by-turn directions.
3. **Easy incremental deployment:** Deploying the system for a new building consists of entering the floor plan into OpenStreetMap.

7.3.1 System Design

Figure 7.4 presents an overview of our system. We obtain map material from OpenStreetMap or from a local source opportunistically, this allows easy updating and incremental deployment on a global scale. After the user selects her route, the accelerometer and compass of the user's phone are used to detect steps and step headings. We then match these steps onto the map using a first fit and a sequence matching scheme. Finally, we present the estimated position back to the user, together with turn-by-turn directions towards the destination.

7.3.1.1 Generating Maps

OpenStreetMap [Ope11b] is an effort to create and distribute free geographic data, such as street maps, but also indoor maps of public buildings, albeit indoor support is still rudimentary [Ope11a]. OpenStreetMap allows wiki-style editing, thereby enabling everyone to contribute easily.

The popularity of OpenStreetMap allows us to make use of a variety of tools, e.g., JOSM [SO11], to create and extend maps incrementally. The OpenStreetMap community has already mapped large parts of the world. At the same time, many companies, such as Microsoft, make satellite imagery available for initial crowd-sourced mapping tasks, which can later be improved by people visiting the area in person. This eases the task to create indoor maps from indoor floor plans the integrate with the outdoor footpaths and streets. To create a map for a new environment, we only need a floor plan which we load into JOSM, align it to the vicinity and vectorize it to create our map data.

7.3.1.2 Step Detection

Modern smartphones are typically equipped with an accelerometer and a compass. These can be directly used for our step detection and step heading estimation. We detect steps from a characteristic regular pattern produced by each contact of the heel with the ground during each step, even when the user holds the phone in her hand in front of herself.

7.3.1.3 Path Matching

Upon detection of a step, we trigger path matching. We propose several strategies for matching detected steps onto expected steps from a map by adapting sequence alignment algorithms [WF74] from the field of bioinformatics, to align the detected steps with the expected steps extracted from a map. Here, the goal is to return the location that results with the best matching of expected steps up to that location and all detected steps.

This is done by calculating all possible matchings of detected steps to the path and returning the location with the least penalty. The penalty for each location indicates how many steps would have to be added, deleted, or changed from the detected steps to match them along the path up to the given location. Thus, the fewer changes are needed, the better the resulting match is. A full algorithmic description and measurement results are available in [BLSVW11].

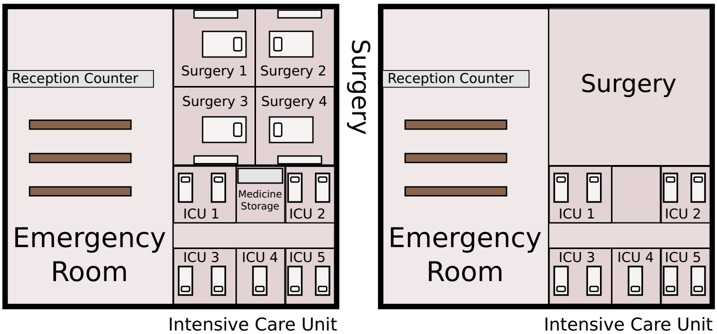


Figure 7.5 Map of a fictional hospital as viewed by two groups with different access privileges: employees and visitors. Left: The master map of the hospital as seen by employees. Right: The filtered map available to visitors. Visitors may need to know the general location of the surgery unit, but they do not need to know the exact layout. Similarly, knowledge about the location of the medicine storage room should not be available to the public. Therefore, this information is filtered.

7.3.2 Map Dissemination

Our system is based on OpenStreetMap. The liberal license of OpenStreetMap allows us to easily store map data for offline use or forward it to other users. However, as the data format and tools are completely open as well, we can also use this information for private mapping information that should only be forwarded to authorized people locally. We explored this idea in the Master thesis of Shahrooz Afsharipour [Afs14].

Our system provides a secure platform for preparing, filtering and disseminating locally stored maps. Filtering of maps allows the creation of different detail levels or of different focus for different users. To implement filtering policies, we make use of MapCSS, a style language similar to Cascading Style Sheets with a focus on maps. By extending it with attributes, such as `remove` and `strip`, we can remove all matching map features from a file or strip selected tags, i.e., annotations, that might be considered confidential. We present an example of such a filtering in Figure 7.5.

After map files are preprocessed in such a way, we can forward them locally opportunistically, or make them available via a local access point or a Bluetooth kiosk. While this use case not necessarily needs opportunistic multi-hop forwarding as is the case of DTNs, interesting questions arise regarding the security of such a system including provable location attestation. Only local authorized users should be able to get the more detailed maps. However, how does a user prove, that she is a certain location and authorized?

7.4 RatPack Project

Moving away from services aimed at users of commodity mobile devices, we now turn towards wireless sensor applications. One of the core motivations for the research in sensor networks is the vision of wide spread deployment in nature to observe environmental phenomena. In the RatPack project, we were investigating the necessary technologies to make this a reality, in particular in the context of the observation of rat behavior in their natural habitat.

Rat behavior in captivity is a well researched area as rats are used as a model organism for various pharmaceutical and cognitive research questions. While we are still far from a comprehensive understanding even in laboratory environments, surprisingly, we know even less about their natural behavior as they dwell in underground burrow systems. Previous research focused mainly on archeology like approaches, digging out existing burrows and determining behavioral patterns based on found remains.

Our approach aimed at eventually giving a more faithful view into the behavior of wild living rats. The basic idea was to equip rats with sensor nodes in leather harnesses similar to a backpack—hence the name: RatPack. These nodes would form a loosely connected network. From this, we planned to derive a time–dependent view of the behavior of rats and patterns previously unobserved or at least unconfirmed in the wild.

After discussing selected related work, we elaborate on the scenario of our application and its impact on the node and software design. Finally, we present approaches, we consider the most appropriate.

7.4.1 Related Work

Over the last 15 years, researcher deployed a considerable number of sensor networks for environmental monitoring [JOW+02a; SPMC04]. Most of these sensor networks, except ZebraNet [JOW+02a], use a static deployment of networks. In these deployments, researchers placed sensor nodes at locations of interest, ensuring that the nodes could communicate with the base station.

The researchers of the ZebraNet project strapped a collar with customized sensor nodes, equipped with a GPS receiver and a solar panel, to the neck of zebras in central Kenya. The nodes recorded the animals' position as well as the time and place of their encounters. From this data, biologists evaluate the zebras' movement and social interactions. Unfortunately, for our goal of observing the behavior of burrowing animals, i.e., rats, GPS is not feasible due to the relative impermissibility of earth to radio signals and its limited accuracy.

A related approach for maritime animal is the DTAG project [JT03] in which Johnson et al. analyze the behavior of whales. Since whales spend most of their time (up to 95%) underwater, radio communication and GPS are infeasible. The researchers developed a special sensor node that attaches to the whale using a suction cup. They track the relative movement under water using accelerometers, a compass and a water pressure sensor and record this data into flash memory. Once the memory is full, the sensor node detaches the recording tag from the whale and floats to the surface where the researchers collect it. Although this approach is very elegant for underwater animals, it is not feasible for research on burrowing animals either.

7.4.2 Requirements and Environmental Constraints

Working with rats puts numerous constraints on the design of sensor nodes. One major limitation is their size: The average adult Norway rat measures 25 cm in length and weighs 250 g [Cal63]. Our sensor nodes needed to be designed in a way that our animals would not be significantly restricted in their natural movements, which strongly limits the available node and battery size.

The main focus of our project was the investigation of the burrowing behavior. We confirmed in experiments with artificial rat burrows that the expected average communication range is 0.9 m along a typical rat burrow tunnel. Through earth, this range can be as low as 0.2 m to 0.3 m. Consequently, these channel characteristics result in a sparsely connected network.

At the same time, memory and communication bandwidth capabilities are also limited by the size and energy constraints of our system. Hence, a trade-off must be found. As we do not expect to know all exits of a rat burrow and some rats may stay in the burrow for long durations, we cannot guarantee that each sensor node will be able to deliver its measurements to the base station before its memory is exhausted.

On the other hand, we could add a high bandwidth radio to the sensor node, as the radio would become too warm during operation while at the same time drawing too much energy from our size constrained batteries.

7.4.3 Routing

The main research focus in the sensor network community is on continuously connected sensor nodes. Thus, although the network topology may vary slightly over time, for example, due to node failure or changing radio conditions, the network infrastructure typically remains stable in these mostly tree-based or beacon-based routing protocols.

In our scenario, a delay tolerant approach [Fal03] seems to be far more practicable. Data would be relayed from one sensor node to another when their bearers, i.e., the rats, meet. It is from this observation, that this thesis was started.

We would place a base station at one (or more) exits of the rat burrow. When a rat would pass along this exit, all measurements, e.g., data collected by this rat as well as the data received from other rats, would be transmitted to this base station.

7.4.3.1 Utility Based Forwarding to Base Station

Data being collected have varying relevance over time, depending on the status of biological insight. For example, in the beginning, the layout of the burrow system might have a higher priority, while later, vocalization information might become more important. The utility of transferring one bundle of data might be determined by various factors such as: the availability of memory on another node, the expected delay of forwarding that data through a specific node to a base station, current energy levels. Depending on the current utility metric, different forwarding decisions can be taken [BLV07].

Figure 7.6 Norway rat equipped with the initial mica2dot based sensor prototype.

7.4.3.2 Social Network Based Forwarding

Social structures in rat societies are expected to follow power-law, as is the case for most animals [JBO03]. This fact can be leveraged if specific nodes need to be addressed. Similar to the famous Milgram experiment, if the receiving node is not known to a forwarding node, that data is forwarded to the node with the highest degree of neighbors, if there is no such node, random walk is used [DH07]. This observation lead to the initial design of SimBetAge in Chapter 5.

7.4.3.3 Data Reduction

At the time of the initial experiments, sensor nodes had very limited storage space, typically 4 kB of RAM and about 500 kB of additional flash space. As discussed, it may take some time until a certain rat passes one of the base stations. Thus, its sensor node needs to store potentially large amounts of measurement data – its own and that of the rats it has (potentially indirectly) met and was chosen to relay. Several ways to reduce the amount of memory necessary exist, among them: lossless compression of the data, using LZW or similar algorithms and lossy source coding schemes.

Our goal was to develop behavioral models which should be validated. Typically, these models require significantly less memory than the original experimental data. Therefore, another focus of our work is not the collection of raw data but rather the automatic derivation of models using techniques from the stream data mining community [GZK05]. A limited amount of supporting evidence and outliers may still be transferred to give researchers additional data for validation. We investigated this data reduction in the context of burrow reconstruction using graph matching [BLFAW10].

7.4.4 Laboratory Prototype

Our first setup consisted of a mica2dot mote powered by a coin cell battery, supporting a number of custom made sensor boards, see Figure 7.6. In our current laboratory prototype each node records data, e.g., data from accelerometer and ultra sound, until its memory is exhausted and subsequently transmits it to the base station when signal quality is sufficiently high.

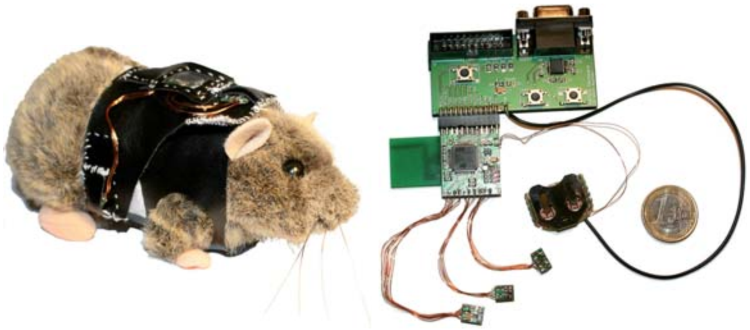


Figure 7.7 RatMote prototype on a plush rat and connected to a programmer. The accelerometer, electronic compass, and gyrometer are connected using flexible cabling to allow flexible placement within the rat's backpack.

This approach is feasible in preliminary test scenarios in the controlled environment of a research center. For use in a real burrow system, however, this approach would need to be adapted for the reasons stated above.

7.4.4.1 Updated hardware

Based on our experience in the initial experiments we designed RatMote, a specialized sensor node [BLBGW10], see Figure 7.7. The core of the RatMote sensor node consists of a STM32 Cortex-M3 processor and a IEEE 802.15.4 compatible radio module. It is equipped with a set of inertial navigation sensors consisting of a 3D accelerometer, a 3D magnetometer and a 2D gyroscope. These sensors allow step detection and path reconstruction even in the absence of anchors or GPS, while the complete setup weighs only 12 g.

The whole sensor node is driven by a reduced supply voltage of 2.7V instead of a more common 3.3V, which leads to a 20 % decrease in power consumption. It also allows us to maintain a stable supply voltage over the whole battery life time of a Lithium-Ion battery without using a switching voltage regulator. This is particularly important for the precise measurement of analog inputs like gyroscopes.

The CPU can run at variable clocking frequencies of up to 64 MHz without an external quartz. We measured its computational efficiency with the Dhrystone and the Whetstone benchmark as well as with the calculation of a FIR filter. We found the most efficient setting of the RatMote CPU for computationally demanding algorithms to be 48MHz. At this setting, RatMote is able to compute 22 times more FIR filter calculations per mAh and is 14 times faster than the widely known TelosB sensor node. Despite a sleeping current of 380μA our platform is still more efficient than a TelosB sensor node, if at least 1.1 FIR filters are calculated per second.

Aside from our project, we envision this new hardware to be useful in a wide range of other research questions, like medical applications, animal tracking, process management, and logistics. The improved processing power allows complex behavior analysis and classification routines.

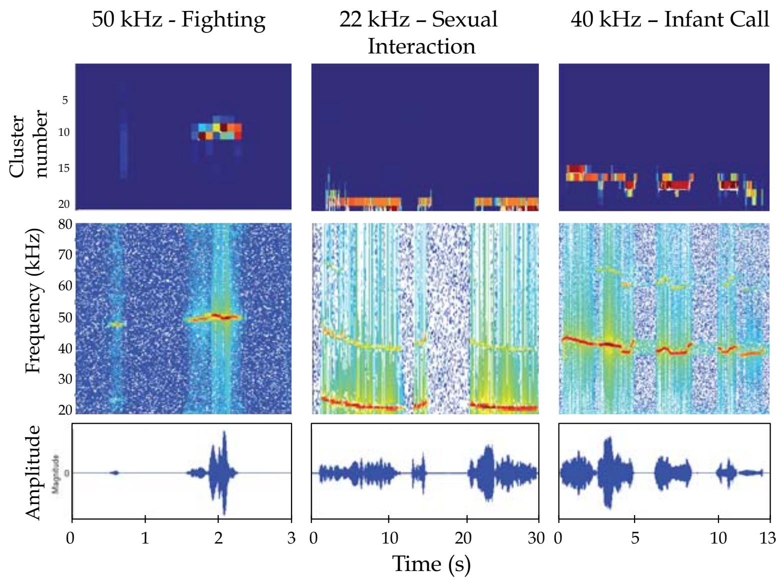


Figure 7.8 Comparison between 3 rat calls that are analyzed by ZCA (upper row), by spectrograms (mid row) or by their amplitude (lower row). The behavioral classification of the calls is following (Voipio, 1997). The results shown here were realized on a test setup running on a mica2dot mote with 10 times delayed playback.

7.4.5 RatPack Results

This section summarizes the vocalization and step detection results achieved within RatPack in collaboration with Johannes Thiele and Okuary Osechas of the University of Tübingen. They were previously published in [TBLOMW10]. Rats live in subterranean burrows in loose assemblies with a varying number of members. They communicate via smell, touch and sound. While the different rat calls are variable, they can be distinguished easily. Also, they map to well-defined internal states of an animal and particular types of interaction between them.

Researchers published a number of studies [Kal90; Voi97] creating a *vocabulary* of vocalization together with behavioral context, such as: resident-intruder, mother-child interactions, or post-ejaculatory and other mating sounds.

The analysis of these sounds allows us to label and classify the interaction between to individual rats meeting, as well as their relationship. So, in addition to a meeting analysis of which tagged rats were sharing a location, this allows us to further classify dominance structures, kinship relations and hierarchies. Figure 7.8 presents a selection of vocalizations we recorded. To enable a low-power micro-controller to classify these, we employed a zero crossings analysis, as a rough approximation of an FFT. As we can see in the figure, this allows an accurate classification.

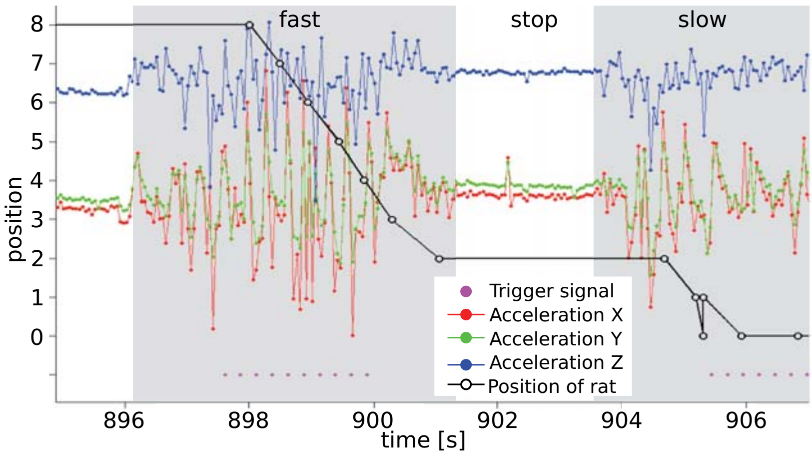


Figure 7.9 Pseudo-step detection. As we cannot put accelerometers on the paws of a rat, we measure the movement and acceleration at the spine. This movement is strongly correlated with walking and allows to distinguish between different speeds and postures.

As GPS and related approaches are not available underground because of limited ground permeability to radio waves, we made use of ego-motion detection using pseudo-step detection, cf. Figure 7.9.

7.4.6 Lessons learned

While the idea of RatPack was easy to grasp on a conceptual level and lended itself very well to demonstrators and outreach projects, it was very ambitious from the start and ultimately, we were not able to complete our vision due to lack of funding and time. However, it provided a starting point for a number of further successful research ideas.

The project combined a number of interwoven challenges, starting from new hardware design with severe energy, weight and size constraints. The complexity of this, we initially underestimated. Going more into the details of underground radio transmission, we quickly learned about the additional communication challenges, leading to the recognition of neighbor discovery as a severely underrated research question.

The planned activity recognition and mapping of the rat burrows resulted in another completely new challenge. While simultaneous localization and mapping (SLAM) is an established research question in the field of robotics, albeit not completely solved yet, the additional inaccuracies of small low power sensors that are carried by rats adds a completely new level of challenges. Addressing these challenges led to new approaches also applicable to human indoor navigation.

7.5 Acknowledgments

DTN applications, in particular the RatPack project, were the initial motivation for this thesis. The idea for the RatPack project developed in 2005 and 2006 during the Master's thesis of the author with significant input from Johannes Thiele, Hanspeter Mallot, Olaf Landsiedel and Klaus Wehrle at the University of Tübingen. We published a first poster on the requirements [BLWOT08] in the SIGCOMM conference in Seattle 2008.

Initial DTN prototypes were developed with the help of Nicolai Viol, who first started as a student assistant in 2007 and later worked on his Diploma Thesis on DTNs [Vio09] in 2008 and 2009 at RWTH Aachen University. The sensor node prototypes were developed together with Okuary Osechas and Johannes Thiele in the context of the RatPack project funded by Microsoft Research. Okuary Osechas and Johannes Thiele were also the driving forces for the biological and physiological measurements of rat vocalization and step detection. The RatMote was developed together with Thomas Bretgeld in the context of his Master thesis in 2010 [Bre10] and published as a demo [BLBGW10] at IPSN 2010. The most comprehensive publication on RatPack is a book chapter [TBLOMW10], to which the author of this thesis contributed mainly on hardware, wireless characterization and routing, while the vocalization and step analysis was mainly contributed by Johannes Thiele and Okuary Osechas.

The ADAMA framework for DTN applications were developed with the help of André Goliath, who worked as a student assistant starting 2009 and later worked on his Diploma Thesis on Quality of Experience as well as the ADAMA framework [Gol12] at RWTH Aachen University. A demo of the framework [GBLW12] was published at ExtremeCom 2012 in the Swiss Alps.

FootPath rose out of the need for an updated map of the computer science center and the idea by Klaus Wehrle of applying the ideas for step detection and graph matching of RatPack to human navigation. In his Bachelor's thesis, Paul Smith developed an initial prototype [Smi11] of the idea which we published at a renown international conference on indoor positioning and indoor navigation (IPIN) [BLSVW11]. Afterwards, we continued to develop these ideas with Paul Smith as a student assistant and in the Master thesis of Felix Gerdsmeier [Ger12b] and the Bachelor thesis of André Pomp [Pom12]. This resulted in another published paper at IPIN 2012 [BLGSW12]. We started to explore the idea of map distribution in the Master thesis of Shahrooz Afsharipour [Afs14].

8

Conclusions and Outlook

In this dissertation, we proposed different approaches as prerequisite to disruption tolerant services. In particular, we identified and characterized services for their feasibility in disruption prone, low or opportunistic connectivity areas. Then, we proposed a new neighbor discovery scheme that allows the discovery of potential communication partners in communication range with minimal latency and minimal energy requirements. Making use of models of human mobility, we propose two new routing schemes based on social network characteristics and on geographical location probability distribution. Finally, we looked at different applications making use of these methods.

In the following, we will summarize the proposed methods, focus on the key insights gained during the writing of this dissertation. Finally, we highlight potential future next steps and research directions.

8.1 Summary

Building successful services for disruption tolerant networking requires three different key steps: (1) identification and characterization of possible applications and services, (2) optimization of neighbor discovery, and (3) efficient routing of data bundles.

- **Identification and characterization of suitable services:** We developed a model for users and services in the context of disruption tolerant networks.
- **Energy-efficient neighbor discovery with minimal latency:** We proposed and evaluated a new scheme for wireless neighbor discovery between unsynchronized nodes based on perfect difference sets.
- **Routing with minimal delay and resource overhead:** We proposed two novel approaches for predicting human mobility using local information for

forwarding messages in disruption tolerant networks: (1) Social Network Based Forwarding, which takes into account, who met whom how often in the past, and (2) Geography Based Forwarding, which works on the frequency at which certain locations were visited in the past.

Based on our contributions in these areas, we looked at particular applications that are particularly well suited for low connectivity environments. In the following, we revisit the specific research questions in each area and summarize our contributions towards usable and efficient services in disruption tolerant networks.

8.1.1 Identification and Characterization of Services

We developed an initial model for users and services in the context of disruption tolerant networks factoring in the service communication structure, together with user experience and user expectations. Where commonly used Quality of Service metrics, e.g., delivery latency and delivery rate, are insufficient to evaluate service acceptance by its users our model aims to capture and predict the resulting outcome for the user, the Quality of Experience.

Accordingly, we identify characteristics of services and networks that are suitable for challenged scenarios. These characteristics include: (1) Data Prefetching and Bundling, (2) Bursty Communication Style, (3) Soft Time Constraints, (4) Lenient Delivery Order Constraints, (5) One-Way Communication, and (6) Hop-to-Hop Usability. We propose new metrics that allow the Quality of Experience evaluation of services: (1) Speed of Interaction, (2) Comfort and Frustration Level, (3) Delivery Effectiveness, and (4) Intercontact Times / Idle Period. While the model is still preliminary, it allows to give guidelines on how to evaluate services and applications.

Finally, we develop a set of sample applications particularly well-suited for challenging network environments. These are a collaborative event scheduling and a Wikipedia access application. We further explored map dissemination for indoor navigation and mobile sensor networks for wildlife monitoring. Through this, we argue that there is no singular killer app for DTN, but rather, we show how considering these challenges to network communication in the design of an application can lead to a universally better experience for all users.

8.1.2 Neighbor Discovery

We proposed and evaluated a new scheme for wireless neighbor discovery between unsynchronized nodes based on perfect difference sets. The construction is a mathematically provable optimal trade-off between the discovery delay and power consumption for discovering the presence of a potential communication partner in range. The exact formulation of the neighbor discovery schedule problem in set theory is based on four properties: (1) The intersection between different schedules cannot be empty. Therefore, a discovery is guaranteed. (2) The schedules should not be trivial, as it would be if the radio were continuously on. (3) All schedules should have equal size. So, each node could in principle the same—potentially shifted—schedule. (4) The schedule should be closed under rotation. The last property ensures that two schedules intersect for all possible phase shifts, therefore allowing unsynchronized devices to discover each other.

The application of Perfect Difference Sets (PDS) allows the provable optimal trade-off. A schedule for a duty cycle of 2% only needs 2451 slots, that is about 4 min assuming 100ms per slot, opposed to 5266 slots, or about 9 min for U-Connect, the next best related work. Generalizing from property 3, we allow the discovery of devices with differing energy restrictions. We prove that Grid is provable fair, in that nodes using more energy will on average discover each other faster. Through an exhaustive search of all possible combination of schedules and phase shifts, we show that PDS also performs fair in practice, that is down to a duty cycle of 1%. However, we were not able to provide a general proof beyond that.

We present a proof of concept implementation of this scheme for the Linux kernel as well as for TinyOS based sensor nodes, which are the basis for further energy efficient local content sharing applications. The Linux kernel implementation is a relatively simple patch to the `mac80211`. This makes it independent of specific wireless cards. In this context, we also elaborate on the design of active radio slots, and how applications can announce new and changed content in an energy efficient way, by using an update counter.

8.1.3 Routing

We proposed two approaches for predicting human mobility using local information for forwarding messages in disruption tolerant networks: (1) Social Network Based Forwarding, which takes into account, who met whom how often in the past, and (2) Geography Based Forwarding, which works on the frequency at which certain locations were visited in the past.

8.1.3.1 Social Network Based Forwarding

Instead of modeling the social interactions in a binary graph, in SimBetAge, we capture the temporal change of a social network in a weighted graph. Edges in the graph degrade over time, if not refreshed. In turn, we extend the definition of similarity and betweenness to capture real-valued edge weights and further propose the *directed ego flow betweenness* as a novel metric to capture the usefulness of a node as a carrier for data bundles.

We evaluated this approach against a number of existing routing schemes, in particular: Direct Delivery, Epidemic Routing, Prophet and the original SimBet. Using the MIT Reality Mining trace, the Haggle imote trace, and the Dartmouth Outdoor Experiment traces, we show that routing performance can be drastically improved when temporal changes are considered.

8.1.3.2 Geography Based Forwarding

While human social interaction is one key heuristic in understanding human mobility, the actual locations visited in the past are also a good predictor on where users will be in the future. In GeoDTN, we modeled human mobility as a probability distribution around known anchor points. This geographic probability function changes over time, just as social interactions do.

In a similar process to the modeling of social ties, we model the probability of locations as degrading over time if not refreshed. Based on this model, we developed a heuristic to predict future connectivity to be used as a connection oracle for disruption tolerant networking based on regional and global mobility.

To be able to evaluate this approach in the absence of widely deployed disruption tolerant consumer applications, we presented traces of self-reported global mobility from the Brightkite website for 221 active users.

The evaluation shows that geoDTN works similarly well as algorithms based on social group characteristics such as SimBetAge and outperforms binary movement algorithms as Move. In dynamic, less periodic networks, geoDTN significantly outperformed the other algorithms in hop count and delivery time. It performs in general 130%, in individual cases even 200%, better in hop count. In terms of delivery time, it outperforms the other algorithms on average by a factor of 3, PRoPHET by factor 6. This demonstrates geoDTN to be a fast self-adapting algorithm for human mobility based networks.

The energy and fairness evaluation shows that geoDTN performs significantly better than the best state of the art algorithm in delivery rate, PRoPHET. GeoDTN needs less hops, thus less transmission energy and distributes the load fairer over the different nodes.

8.2 Lessons Learnt

After having summarized the contributions of this work, we will take a step back and highlight again the underlying key findings. These observations served as a starting point for this work. Inherent to these observations, however, come corresponding limitations.

8.2.1 There is no singular Killer App for DTNs

While the community is on a quest for the DTN killer app, we argue that most if not all mobile applications would highly benefit from becoming aware of delays and disruptions, and varying connectivity over time.

Also, there are no universally useful evaluation and usability metrics for communication protocols. While we rate today's protocols mainly based on throughput, hop count, and delivery time, these numerical measures only capture limited aspects of performance. In challenged environments, they are only of limited use, as they are dominated by the network and connectivity itself. Therefore, we need fundamentally different metrics to judge the utility of emerging protocols for challenged networks.

The absence of a singular killer app makes it more difficult to convince application developers to support challenged networks as they do not seem strictly necessary. Adoption of DTN protocols will therefore be slow and gradual.

8.2.2 Cyclic Block Designs are Useful for Neighbor Discovery

The subfield of block design from the field of combinatorics is typically used for experimental design. It represents a family of related subsets of a set of elements that fulfill specific properties. In particular, perfect difference sets, where each subset of a specific block design has exactly one element in common with all other subsets of the block design, fit the problem of neighbor discovery: For each pair of communication nodes, the radio activation schedule needs to guarantee that they are active at the same time at least once during a discovery period.

In particular, perfect difference sets are provably the smallest sets of radio activation schedules that fulfill all neighbor discovery requirements. However, the make up of a perfect difference sets are not trivial to predict, so general statements beyond its existence and size are difficult, requiring exhaustive search for absolute statements.

8.2.3 People Meet People They Met Before

The assumed mobility patterns for most Mobile Ad-hoc Network (MANET) protocols are various forms of random walks, ignoring social and geographical aspects of human mobility. Human mobility incorporates a number of characteristics: Friendship, familiarity and employment aspects guide who we meet how regularly. Humans are territorial. The likelihood of coming back to a place, we visited in the past is much higher than the likelihood of returning to a place we never visited. Routing schemes taking these characteristics into account may have the potential to significantly improve routing performance.

To make use of these aspects of human mobility comes at the price of processing overhead and possible privacy violations. To decide who is the better carrier of a message, each potential carrier has to reveal all her social contacts or all the places she visits. While this significantly improves routing performance, this point alone may make the deployment of derived routing schemes prohibitive for a wider or even global deployment.

8.3 Future Work

Over the course of this work, as is the case for most research, the answer to a given initial questions raises up and gives rise to a plethora of new ideas and research questions. In the following, we highlight a selection of possible future directions for this work.

8.3.1 Quality of Experience

Quality of experience is an interdisciplinary research field, and cannot be viewed purely through the lens of protocol engineering. To properly be able to judge the effect of a particular design decisions, the role of psychology, sociology, but also user interface design cannot be understated.

As such, we are only at the beginning of a more comprehensive understanding of Quality of Experience, requiring extensive collaborations between all involved fields to define metrics, perform user studies and understand the causality between the effects.

8.3.1.1 Larger QoE User Studies

Our initial user tests with only 12 participants are mere feasibility tests that may only inform future research directions and indicate fundamental problems. However, on a principal level, they are too small for authoritative statements. We need a significantly larger sample size with participants from different technical backgrounds and socioeconomic groups to properly validate these types of models.

8.3.1.2 Better QoE Metrics and Thresholds

We propose new measures for the context of Quality of Experience evaluation of services: (1) Speed of Interaction, (2) Comfort and Frustration Level, (3) Delivery Effectiveness, and (4) Intercontact Times / Idle Period. However, what are acceptable thresholds for each of these measures? The thresholds still depend heavily on the actual application and use case and more general quantitative statements are not yet possible.

8.3.2 Neighbor Discovery

Neighbor discovery is of tremendous importance for mobile devices. Neighbor discovery is the main mode of operation in challenging network environments. As such, optimizing energy usage of neighbor discovery has a significant effect on the battery life time of a mobile device. Still, the topic only finds its way slowly into protocol standardization with many research questions still wide open.

8.3.2.1 Synchronous neighbor discovery

In our approach, we were focusing on *asynchronous* neighbor discovery, that is we did not require any time synchronization between nodes. However, in particular for mobile devices such as mobile phones that have an integrated GPS receiver, this requirement might actually not be necessary.

The GPS signal includes a reference time with an accuracy of 10 ns based on atomic clocks. If each network node synchronizes itself occasionally with the GPS signal to avoid clock drift, a whole new set of discovery mechanism becomes possible. Each node may turn on its radio every minute for 100 ms. This would lead to a maximum discovery latency of 1 minute at a duty cycle of under 0.2 %, a value deemed impossible for unsynchronized schemes.

Specific research questions include: What is the clock drift in mobile devices? How short can we make the time that devices have to turn on their radio? Which discovery latencies make sense for which use cases? What offset of the discovery slot to, e.g., the beginning of a minute should we choose? Why? How can asynchronous and synchronous networks interact?

8.3.2.2 Discovery on Multiple Frequencies

Currently all nodes are assumed to operate on the same well-defined frequency. If we relax this requirement and allow different frequencies, possible in the context of cognitive radio, where our application is only a secondary user of a specific frequency, this problem becomes potentially an order of magnitude more difficult.

Specific research questions include: How do we model the multi-frequency problem as a quorum design? Will the primary user of a frequency affect both potential secondary users trying to detect each other equally? If no, can we use this to our advantage?

8.3.2.3 Discovery using Multiple Radios

A similar question arises, if devices feature multiple radio interfaces, e.g., several independent Wi-Fi interfaces, or a Bluetooth and a Wi-Fi interface. Current mobile devices are already equipped with the latter. However, Bluetooth and Wi-Fi currently are handled by the same integrated circuit and share their antenna.

Specific research questions include: Can both interface can be truly used independently, or is there crosstalk between the different functions that interferes? Should one interface serve as the primary discovery radio or are there energy efficient schedules that allow both interfaces concurrently? Do we need a multi radio MAC layer or at least multi radio MAC coordination?

8.3.2.4 Application Programming Interfaces for Neighbor Discovery

In our discovery scheme, we proposed the use of an *update counter*, to signal potential communication partners, if we have new information. Bluetooth Low Energy, which has only be standardized during the course of this thesis, allows application payload in its advertisement packets. Using interfaces, such as the `AdvertiseData` and the `AdvertiseSettings` interface in Android allow applications to control some of its parameters. However, many questions in the design of the API for Discovery remain open.

Specific research questions include: What are the privacy implications of an update counter? Is an update counter enough? What are the privacy implications of neighbor discovery? How can we counteract specific tracking attacks?

8.3.3 DTN Routing

Over the last decade, a significant number of routing protocols for DTNs have been suggested. Still, due to the lack of a singular killer application for DTNs, none has yet come ahead of the others, but most actually ended up being described in a single paper and never heard of again. In this dissertation, we proposed two new routing schemes based on human mobility models, that are potentially precluded from wider usage due to privacy implications. To counter this admittedly important argument, several questions remain open.

8.3.3.1 Privacy in DTN Routing

A network participant does not necessarily want to publicly reveal all her social contact or each location she has ever visited how often. This is a very serious argument against this type of routing schemes that needs to be addressed.

Specific research questions include: How can we make useful routing decisions without compromising a user's privacy? How could pseudonyms and the regular change of pseudonyms help?

8.3.3.2 Reputation and Micropayments for DTN Services

Now, if a user needs to reveal personal information to communication partners to allow efficient routing, the question of who to trust with that information, and potentially, could I pay for a specific service or enforce a punishment for misbehavior becomes relevant.

Specific research questions include: How can we maintain trust and reputation in a disconnected setting? If we allow occasional global connectivity, can we resolve this challenge more efficiently? Do we need a particular trusted third party or are fully distributed mechanisms like a block chain applicable? How can a node prove misbehavior of other nodes? How can a node pay another node for forwarding?

8.3.3.3 Name Resolution in DTNs

While DTNs are designed to use late binding of names, many routing schemes still require specific information, such as a geographical location probability to work. The question how to realize this look up is still mostly open.

Specific research questions include: How can we translate a particular name to a geographical location in the network? What are the privacy implications of this approach? Does the assumption of always routing from or towards the globally connected internet help? Would a network entities in the globally connected internet under the control of the particular mobile disconnected user help?

8.3.4 DTN Aware Applications and Services

As we repeatedly claim, the majority of applications and services would benefit from a better awareness of the network context of the user. However, for this to become a reality, we need proper interfaces that make applications aware of the network context and allow the application to properly signal its communication intentions to the network stack. In the following, we explore a range of possible directions.

8.3.4.1 Context Awareness in Services

While especially mobile operating systems allow applications to query the current networking context, the current location, and user activity, semantic context information is often missing. What does a particular location signify to a user? Similarly, the opposite direction of signaling is still largely missing. Applications cannot easily signal to the operating system that a particular stream will be high throughput but short, or long running but with only minimal traffic but requires high reliability.

Specific research questions include: How can an application signal to the network stack its expected future behavior? As specific file systems allow applications to hint how they will access a file using `madvise`, how would an API for the network stack look like?

8.3.4.2 Context Prediction for DTN Aware Services

While interfaces to query the current context of the user exist, no such interface exist yet, that signal near future context. If an application was aware that a user is about to leave a connected area in the next minute, it may sync its state, which it may have otherwise deferred to a later time.

Specific research questions include: How can the system signal expected future connectivity? How much control should be given to a specific application, or should an application only install hooks for certain well defined conditions? How certain should the operating system to signal a prediction?

8.3.4.3 Disruption Tolerant Security

While the previous directions were local to a particular device, the question of security in disruption prone, low connectivity areas addresses the whole network. Key and identity management are largely open in this domain.

Specific research questions include: How to deal with key distribution and key revocation in networks without global connectivity? How can we manage reputation without a trusted third party that is continuously reachable by all participants?

Bibliography

Coauthored publications

[AKBLWW10] Muhammad Hamad Alizai, Bernhard Kirchen, Jó Agila Bitsch Link, Hanno Wirtz, and Klaus Wehrle. "Poster Abstract: TinyOS Meets Wireless Mesh Networks". In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys 2010), Zurich, Switzerland*. New York, NY, USA: ACM, Nov. 2010, pp. 429–430.

[ALBLGW09a] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Agila Bitsch Link, Stefan Götz, and Klaus Wehrle. "Bursty Traffic over Bursty Links". In: *Proceeding of 7th ACM Conference on Embedded Networked Sensor Systems (Sensys 09), Berkeley, CA, USA*. New York, NY, USA: ACM, Nov. 2009, pp. 71–84.

[ALBLGW09b] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Agila Bitsch Link, Stefan Götz, and Klaus Wehrle. *Routing Over Bursty Wireless Links*. Tech. rep. Technical University Hamburg, Sept. 2009, pp. 63–66.

[AVL+11] Muhammad Hamad Alizai, Tobias Vaegs, Olaf Landsiedel, et al. "Probabilistic Addressing: Stable Addresses in Unstable Wireless Networks". In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2011), Chicago, IL, USA*. New York, NY, USA: ACM, Apr. 2011.

[BCZBLW13] Marcel Bosling, Matteo Ceriotti, Torsten Zimmermann, Jó Agila Bitsch Link, and Klaus Wehrle. "Fingerprinting Channel Dynamics in Indoor Low-Power Wireless Networks". In: *Proceedings of the 8th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2013), Miami, USA*. ACM, Sept. 2013, pp. 65–72.

[BLAW13] Jó Agila Bitsch Link, Alaa Alhamoud, and Klaus Wehrle. "A Global One-Way Control Channel for Opportunistic Networks". In: *Proceedings of the 5th Extreme Conference on Communication*. Ed. by Pan Hui and Anders Lindgren. ACM, Aug. 2013, pp. 1–6.

[BLBGW10] Jó Agila Bitsch Link, Thomas Bretgeld, André Goliath, and Klaus Wehrle. "Demo Abstract: RatMote - A Sensor Platform for Animal Habitat Monitoring". In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2010), Stockholm, Sweden*. IPSN '10. Demo Abstract. ACM. New York City, NY, USA: ACM, Apr. 2010, pp. 432–433.

[BLFAW10] Jó Ágila Bitsch Link, Gregor Fabritius, Muhammad Hamad Alizai, and Klaus Wehrle. "BurrowView - Seeing the world through the eyes of rats". In: *Proceedings of the Second IEEE International Workshop on Information Quality and Quality of Service for Pervasive Computing*. IQ2S 2010. (Mannheim, Germany, Mar. 29–Apr. 2, 2010). 2010, pp. 56 –61.

[BLGSW12] Jó Agila Bitsch Link, Felix Gerdsmeier, Paul Smith, and Klaus Wehrle. "Indoor Navigation on Wheels (and on Foot) using Smartphones". In: *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 11-13 November 2012 Sydney, Australia*. IEEE. Piscataway, NJ, USA, Nov. 2012, 10 S.

[BLRIFCW15] Jó Agila Bitsch Link, Roann Ramos, Tim Ix, Paula Glenda Ferrer Cheng, and Klaus Wehrle. "Psychologist in a Pocket: Towards Depression Screening on Mobile Phones". In: *Proceedings of the 12th International Conference on Wearable Micro and Nano Technologies for Personalized Health*. Ed. by Bernd Blobel, Maria Lindén, and Mobyen Uddin Ahmed. Vol. 211. Studies in Health Technology and Informatics. Amsterdam: IOS Press, June 2015, pp. 153 –159.

[BLROW14] Jó Agila Bitsch Link, Roann Ramos, David Orlea, and Klaus Wehrle. "Demo: Opportunistic Mobile Brain Research". In: *Proceedings of the 6th Extreme Conference on Communication*. Ed. by Pan Hui and Anders Lindgren. ACM, Aug. 2014, pp. 1–2.

[BLRSW15] Jó Agila Bitsch Link, Roann Ramos, Cassandra Severijns, and Klaus Wehrle. "Towards Brain Research in a Pocket: Bringing EEG Research and Diagnostics out of the Lab". In: *Proceedings of the 12th International Conference on Wearable Micro and Nano Technologies for Personalized Health*. Ed. by Bernd Blobel, Maria Lindén, and Mobyen Uddin Ahmed. Vol. 211. Studies in Health Technology and Informatics. Amsterdam: IOS Press, June 2015, pp. 185 –190.

[BLSVW11] Jó Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle. "FootPath: Accurate Map-based Indoor Navigation Using Smartphones". In: *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes, Portugal*. IEEE, Sept. 2011, pp. 1–8.

[BLSVW12] Jó Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle. "Accurate Map-based Indoor Navigation on the Mobile". In: *Journal of Location Based Services* 7.1 (July 2012), pp. 23–43.

[BLSW11a] Jó Agila Bitsch Link, Daniel Schmitz, and Klaus Wehrle. "geoDTN: Geographic Routing in Disruption Tolerant Networks". In: *IEEE GLOBECOM 2011 - Next Generation Networking Symposium (GC'11 - NGN), Houston, Texas, USA*. IEEE. Piscataway, NJ, USA, Dec. 2011, pp. 1 –5.

[BLSW11b] Jó Agila Bitsch Link, Paul Smith, and Klaus Wehrle. "FootPath: Accurate Map-based Indoor Navigation Using Smartphones (short paper)". In: *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes, Portugal.* short paper (long paper is also available). IEEE, Sept. 2011, pp. 1–4.

[BLTW14] Jó Agila Bitsch Link, Angel Tchorbadjiiski, and Klaus Wehrle. "WIP: Opportunistic Vote Delegation for e-Voting based on Liquid Democracy". In: *Proceedings of the 6th Extreme Conference on Communication.* Ed. by Pan Hui and Anders Lindgren. Work in Progress Paper. ACM, Aug. 2014, pp. 1–4.

[BLVGW09a] Jó Agila Bitsch Link, Nicolai Viol, André Goliath, and Klaus Wehrle. "SimBetAge: Dealing with Change in Social Networks for Pocket Switched Networks". In: *Proceedings of the 1st ACM workshop on User-provided networking: challenges and opportunities, Rome, Italy.* Ed. by Oliver Marcé Paulo Mendes. U-NET '09. ACM. New York City, NY, USA: ACM, Dec. 2009, pp. 13–18.

[BLVGW09b] Jó Ágila Bitsch Link, Nicolai Viol, André Goliath, and Klaus Wehrle. "SimBetAge: Utilizing Temporal Changes in Social Networks for Delay/Disconnection Tolerant Networking". In: *Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2009), Toronto, ON, Canada.* MobiQuitous 2009. (Toronto, ON, Canada, July 13–16, 2009). Poster and Abstract. 2009.

[BLWOT08] Jó Ágila Bitsch Link, Klaus Wehrle, Okuary Osechas, and Johannes Thiele. "RatPack – Communication in a Sparse Dynamic Network (Poster Abstract)". In: *Proceedings of the ACM SIGCOMM 2008 conference on Data communication.* SIGCOMM 2008. (Seattle, WA, USA, Aug. 17–22, 2008). Poster Abstract. 2008, pp. 467–468.

[BLWOTM07] Jó Agila Bitsch Link, Klaus Wehrle, Okuary Osechas, Johannes Thiele, and Hanspeter Mallot. "Ratpack: Using Sensor Networks for Animal Observation". In: *6th GI/ITG KuVS Fachgespräch "Wireless Sensor Networks", Aachen, Germany.* Ed. by Klaus Wehrle. Vol. 2007-11. AIB. GI/ITG Fachgruppe "Kommunikation und Verteilte Systeme". Aachen, Germany: RWTH Aachen University, July 2007, pp. 95 –97.

[BLWSW11] Jó Agila Bitsch Link, Christoph Wollgarten, Stefan Schupp, and Klaus Wehrle. "Perfect Difference Sets for Neighbor Discovery". In: *Proceedings of the 3rd Extreme Conference of Communication (ExtremeCom 2011), Manaus, Brazil.* Ed. by Pan Hui and Anders Lindgren. ACM, Sept. 2011, pp. 1–6.

[GBLW12] André Goliath, Jó Ágila Bitsch Link, and Klaus Wehrle. "A Versatile Architecture for DTN Services (Demo)". In: *Proceedings of the 4th Extreme Conference on Communication*. ExtremeCom 2012. (Zurich, Switzerland, Mar. 10–14, 2012). Ed. by Pan Hui and Anders Lindgren. 2012, pp. 1–2.

[HPBLG15] Oliver Hohlfeld, André Pomp, Jó Agila Bitsch Link, and Dennis Guse. "On the Applicability of Computer Vision based Gaze Tracking in Mobile Scenarios". In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. New York, NY, USA: ACM, Aug. 2015, pp. 1–8.

[LBLDW06] Olaf Landsiedel, Jó Agila Bitsch Link, Katharina Denkinger, and Klaus Wehrle. "Modular Communication Protocols for Sensor Networks". In: *European Workshop on Wireless Sensor Networks (EWSN 2006), Zurich Switzerland*. Ed. by Kay Römer, Holger Karl, and Friedemann Matterns. Vol. 507. Technical Report ETH Zurich / Dept. of Computer Science. (Poster and Abstract). EWSN. Zurich, Switzerland: Department of Computer Science, ETH Zurich, Feb. 2006, pp. 22 –23.

[LBLW11] Hendrik vom Lehn, Jó Ágila Bitsch Link, and Klaus Wehrle. "Work in Progress: Uncovering the Privacy Implications of Web Usage [Poster]". In: *8th USENIX Symposium on Networked Systems Design and Implementation*. NSDI '11. (Boston, MA, USA, Mar. 30–Apr. 1, 2011). poster and abstract. 2011.

[LBLWTM06] Olaf Landsiedel, Jó Ágila Bitsch Link, Klaus Wehrle, Johannes Thiele, and Hanspeter Mallot. "Rat Watch: Using Sensor Networks for Animal Observation". In: *ACM Workshop on Real-World Wireless Sensor Networks (RealWSN) in conjunction with ACM MobiSys*. (Uppsala, Sweden). Ed. by Pedro José Marron and Thiemo Voigt. Vol. 1. (Poster and Abstract). ACM. SICS, 2006, pp. 1–2.

[LFSBLMA14] Cheng Li, Tomas Fernandez-Steeger, Jó Agila Bitsch Link, Matthias May, and Rafig Azzam. "Use of MEMS accelerometers/inclinometers as a geotechnical monitoring method for ground subsidence". In: *Acta Geodynamica et Geomaterialia* 11.4 (Oct. 2014), pp. 1–12.

[LGPBLW12] Mónica Alejandra Lora Girón, Alexander Paulus, Jó Agila Bitsch Link, and Klaus Wehrle. "Gossipmule: Scanning and Disseminating Information Between Stations in Cooperative WLANs (Poster)". In: *Proceedings of the Workshop on Mobile Opportunistic Networking ACM/SIGMOBILE MobiOpp 2012, Zurich, Switzerland*. (Poster). New York, NY, USA: ACM, Mar. 2012, pp. 87–88.

[PFS+12] Michael Peter, Dieter Fritsch, Bernhardt Schäfer, et al. "Versatile Geo-referenced Maps for Indoor Navigation of Pedestrians". In: *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia*. Ed. by Binghao Li Li and Thomas Gallagher. School of Surveying and Geospatial Engineering, University of New South Wales, Sydney, Australia, Nov. 2012, pp. 1–4.

[SBLAW08a] Raimondas Sasnauskas, Jó Agila Bitsch Link, Muhammad Hamad Alizai, and Klaus Wehrle. *Bug Hunting in Sensor Network Applications*. Tech. rep. Berlin, Germany: Chair of Communication and Distributed Systems (ComSys), Sept. 2008.

[SBLW13] Paul Smith, Jó Agila Bitsch Link, and Klaus Wehrle. "Demo: Opportunistic Deployment Support for Wireless Sensor Networks". In: *Proceedings of the 5th Extreme Conference on Communication*. Ed. by Pan Hui and Anders Lindgren. won "coolest demo award". ACM, Aug. 2013, pp. 1–2.

[TBLOMW10] Johannes Thiele, Jó Ágila Bitsch Link, Okuary Osechas, Hanspeter Mallot, and Klaus Wehrle. "Dynamic Wireless Sensor Networks for Animal Behavior Research". In: *Recent Advances in Biomedical Engineering*. Ed. by Domenico Campolo. Vienna, Austria: InTech, Jan. 2010. Chap. 32, pp. 629–644.

[TOBLW08] Johannes Thiele, Okuary Osechas, Jó Agila Bitsch Link, and Klaus Wehrle. "Smart Sensors for Small Rodent Observation". In: *Proceedings of IEEE Sensors 2008, Lecce, Italy*. IEEE. New York City, NY, USA: IEEE, Oct. 2008, pp. 709 –711.

[VABLW11] Tobias Vaegs, Muhammad Hamad Alizai, Jó Ágila Bitsch Link, and Klaus Wehrle. "Efficient Power Management Using Out-of-Band Signaling". In: *Proceedings of the 10th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*. FGSN '11. (Paderborn, Germany, Sept. 15–16, 2011). 2011, pp. 77–80.

[VBLWRW12] Nicolai Viol, Jó Agila Bitsch Link, Hanno Wirtz, Dirk Rothe, and Klaus Wehrle. "Hidden Markov Model-based 3D Path-matching using Raytracing-generated Wi-Fi Models". In: *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia*. Nov. 2012, pp. 1–10.

Coadvised Theses

[Afs14] Shahrooz Afsharipour. "MapDispenser – A Secure Distribution Mechanism for (Indoor) Navigation Maps". Master's Thesis. RWTH Aachen University, Sept. 2014.

[Akc08] Eldar Akchurin. "Cognitive Radio Management for Wireless Sensor Networks". external thesis with microsoft. Master's Thesis. RWTH Aachen University, Oct. 2008.

[Alh12] Alaa Alhamoud. "A Global One-Way Control Channel For Opportunistic Networks". Master's Thesis. RWTH Aachen University, Mar. 2012.

[Bau09] Patrick Baum. "Stream Data Mining in Wireless Sensor Networks". Diploma Thesis. RWTH Aachen University, Aug. 2009.

[Bra11] Lucas Braun. "TinyOS for early landslide detection". Bachelor's Thesis. RWTH Aachen University, Nov. 2011.

[Bre10] Thomas Bretgeld. "RatMote – A New Wireless Sensor Platform for a Subterranean Animal Habitat Observation System". Diploma Thesis. RWTH Aachen University, Jan. 2010.

[Dat08] Avijit Datta. "Support of Real-time Applications by IEEE 802.15.4-based Systems". external thesis with philips. Master's Thesis. RWTH Aachen University, Feb. 2008.

[Fab09] Gregor Bernhard Fabritius. "BurrowView – Path Reconstruction and Localization in a Subterranean Animal Habitat Observation System". Diploma Thesis. RWTH Aachen University, Sept. 2009.

[Ger12a] Levin Gerdes. "WSNs for Wind Generator Structural Stability". Bachelor's Thesis. RWTH Aachen University, Nov. 2012.

[Ger12b] Felix Gerdsmeier. "Smartphone-Based Multipath Indoor Navigation". Diploma Thesis. RWTH Aachen University, Aug. 2012.

[Gol12] André Björn David Goliath. "ADAMA – An Architecture for Disruption-Aware Mobile Applications". Diploma Thesis. RWTH Aachen University, Apr. 2012.

[Jan12] Sebastian Jansen. "Privacy-aware Application Migration". Diploma Thesis. RWTH Aachen University, Apr. 2012.

[May10] Matthias May. "RatMote2 – Improved Radio and Sensor Performance for an Subterranean Animal Habitat Observation System". Bachelor's Thesis. RWTH Aachen University, July 2010.

[Ngu08] Thanh Phuong Nguyen. "Data Classification in Wireless Sensor Network for Habitat Monitoring". Master's Thesis. RWTH Aachen University, Sept. 2008.

[Pom12] André Pomp. "Smartphone-Based Indoor Mapping". Bachelor's Thesis. RWTH Aachen University, Mar. 2012.

[Sch10a] Daniel Schmitz. "geoDTN – Geographic Routing in Delay Tolerant Networks". Diploma Thesis. RWTH Aachen University, Oct. 2010.

[Sch10b] Stefan Schupp. "Using Quorum Schemes for Neighbor Discovery in Wireless Sensor Networking". Bachelor's Thesis. RWTH Aachen University, Nov. 2010.

[Smi11] Paul Anthony Smith. "Tracking Progress on Path for Indoor Navigation". Bachelor's Thesis. RWTH Aachen University, Apr. 2011.

[Tch12] Angel Tchorbadjiiski. "Liquid Democracy". Diploma Thesis. RWTH Aachen University, Mar. 2012.

[Vio09] Nicolai Lune Viol. "SimBetAge – Design and Evaluation of Efficient Delay Tolerant Routing in Mobile WSNs". Diploma Thesis. RWTH Aachen University, May 2009.

[Wol11] Christoph Wollgarten. "Energy Efficient Neighbor Discovery for Podcast Sharing on Mobile Devices under Linux". Diploma Thesis. RWTH Aachen University, June 2011.

[Wu09] Qian Wu. "Neighbor Discovery in Wireless Sensor Networks". Master's Thesis. RWTH Aachen University, Oct. 2009.

[Wu12] Marc Wu. “Extending iOS: Support for Application Mobility”. coadvised with Uta Christoph. Diploma Thesis. RWTH Aachen University, May 2012.

General Bibliography

[ASCS10] Mohammed Al-Siyabi, Haitham Cruickshank, and Zhili Sun. “Quality of service provisioning for delay tolerant network by implementing admission control model for aircrafts bundles data transmission”. In: *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. IWCMC ’10. Caen, France: ACM, 2010, pp. 706–710. ISBN: 978-1-4503-0062-9.

[ASCS11] Mohammed Al-Siyabi, Haitham Cruickshank, and Zhili Sun. “DTN QoS metrics and fair resources management model”. In: *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*. 2011, pp. 000704 –000707.

[BGJL06a] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks”. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. 2006, pp. 1 –11.

[BHG06] D. Brockmann, L. Hufnagel, and T. Geisel. “The scaling laws of human travel”. In: *Nature* 439.7075 (2006), pp. 462–465.

[BHT+03] S. Burleigh, A. Hooke, L. Torgerson, et al. “Delay-tolerant networking: an approach to interplanetary Internet”. In: *Communications Magazine, IEEE* 41.6 (2003), pp. 128 –136. ISSN: 0163-6804.

[BLM+08] John Burgess, Brian Neil Levine, Ratul Mahajan, et al. *CRAWDAD data set umass/diesel (v. 2008-09-14)*. Downloaded from `http://crawdad.cs.dartmouth.edu/umass/diesel`. Sept. 2008.

[BLV07] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. “DTN routing as a resource allocation problem”. In: *SIGCOMM Comput. Commun. Rev.* 37.4 (2007), pp. 373–384.

[Bur07] Scott Burleigh. “Interplanetary overlay network: An implementation of the DTN bundle protocol”. In: *2007 4th IEEE Consumer Communications and Networking Conference*. IEEE. 2007, pp. 222–226.

[Cal63] J. Calhoun. *The Ecology and Sociology of the Norway Rat*. U.S. Dept. of Health, Education, and Welfare, Public Health Service, 1963.

[CD06] C.J. Colbourn and J.H. Dinitz. *Handbook of combinatorial designs*. CRC press, 2006. ISBN: 1584885068.

[CHC+05] A. Chaintreau, P. Hui, J. Crowcroft, et al. *Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding*. Tech. rep. UCAM-CL-TR-617. University of Cambridge, Computer Laboratory, 2005.

[DC08a] Prabal Dutta and David Culler. "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications". In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. SenSys '08. Raleigh, NC, USA, 2008, pp. 71–84. ISBN: 978-1-59593-990-6.

[DD10] Elwyn Davies and Avri Doria. *Functional Specification for DTN Infrastructure Software*. Tech. rep. N4C, 2010. URL: `http://www.n4c.eu/Download/n4c-wp2-023-dtn-infrastructure-fs-12.pdf` (visited on 08/05/2016).

[Dem08] M. Demmer. "DTNServ: A case for service classes in Delay Tolerant Networks". In: *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on.* 2008, pp. 177 –184.

[DFGV03] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages". In: *MobiHoc '03*. New York, NY, USA: ACM Press, 2003.

[DH07] Elizabeth M. Daly and Mads Haahr. "Social network analysis for routing in disconnected delay-tolerant MANETs". In: *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. MobiHoc '07. Montreal, Quebec, Canada: ACM, 2007, pp. 32–40. ISBN: 978-1-59593-684-4.

[EB05] Martin Everett and Stephen P. Borgatti. "Ego network betweenness". In: *Social Networks* 27.1 (2005), pp. 31–38.

[EB15] Mats Eriksson and Jaap van de Beek. "Is Anyone Out There? 5G, Rural Coverage and the next 1 billion". In: *IEEE ComSoc Technology News* (2015). URL: `http://www.comsoc.org/ctn/anyone-out-there-5g-rural-coverage-and-next-1-billion`.

[EP06] Nathan Eagle and Alex S. Pentland. "Reality Mining: Sensing Complex Social Systems". In: *Personal Ubiquitous Comput.* 10.4 (May 2006), pp. 255–268. ISSN: 1617-4909.

[Fal03] Kevin Fall. "A delay-tolerant network architecture for challenged internets". In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM '03. Karlsruhe, Germany: ACM, 2003, pp. 27–34. ISBN: 1-58113-735-4.

[FBW91] Linton C. Freeman, Stephen P. Borgatti, and Douglas R. White. "Centrality in valued graphs: A measure of betweenness based on network flow". In: *Social Networks* 13.2 (1991).

[FC06] Stephen Farrell and Vinny Cahill. *Delay- and Disruption-Tolerant Networking*. Artech House Publishers, 2006.

[FF56] L.R. Ford and D.R. Fulkerson. "Maximal flow through a network". In: *Canadian Journal of Mathematics* 8.3 (1956), pp. 399–404.

[FMM+11] Stephen Farrell, Alex McMahon, Eoin Meehan, et al. *Report on an Arctic Summer DTN 2010 Trial*. Tech. rep. DRAFT 2011-05-18 Work-in-progress. Trinity College, Dublin, Ireland, 2011. URL: http://dtn.dsg.cs.tcd.ie/n4c-summer10/summer10.pdf.

[GKN+] Robert S. Gray, David Kotz, Calvin Newport, et al. *CRAWDAD data set dartmouth/outdoor (v. 2006-11-06)*. Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/outdoor. URL: http://crawdad.cs.dartmouth.edu/dartmouth/outdoor.

[GKN+04] Robert S. Gray, David Kotz, Calvin Newport, et al. "Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms". In: *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWiM '04. Venice, Italy: ACM, 2004, pp. 220–229. ISBN: 1-58113-953-5.

[Gre07] Andy Green. *Penumbra, Broadcast-based Wireless network*. website defunct, only available at: https://web.archive.org/web/20070128135811/http://penumbra.warmcat.com/_twk/tiki-index.php. 2007. URL: http://penumbra.warmcat.com/.

[GZK05] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. "Mining data streams: a review". In: *ACM Sigmod Record* 34.2 (2005), pp. 18–26.

[HCG+06] P. Hui, A. Chaintreau, R. Gass, et al. "Pocket Switched Networking: Challenges, feasibility and implementation issues". In: *LNCS* 3854 (2006), p. 1.

[HCS+05] Pan Hui, Augustin Chaintreau, James Scott, et al. "Pocket Switched Networks and Human Mobility in Conference Environments". In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*. WDTN '05. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 244–251. ISBN: 1-59593-026-4.

[HCS+11] Theus Hossmann, Paolo Carta, Dominik Schatzmann, et al. "Twitter in disaster mode: security architecture". In: *Proceedings of the Special Workshop on Internet and Disasters*. SWID '11. Tokyo, Japan: ACM, 2011, 7:1–7:8. ISBN: 978-1-4503-1044-4.

[HCY08] Pan Hui, Jon Crowcroft, and Eiko Yoneki. "Bubble rap: social-based forwarding in delay tolerant networks". In: *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. MobiHoc '08. Hong Kong, Hong Kong, China: ACM, 2008, pp. 241–250. ISBN: 978-1-60558-073-9.

[HT06] Harri Holma and Antti Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. John Wiley & Sons, 2006.

[HYKPK10] O. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson. "A mobile peer-to-peer system for opportunistic content-centric networking". In: *MobiHeld*. Sept. 2010.

[IEE07] IEEE. *IEEE Standard for Information technology, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. 2007.

[IP09] Rahim Tafazolli Ioannis Psaras Ning Wang. “Six years since first DTN papers: Is there a clear target?” In: *1st Extreme Workshop on Communication*. ExtremeCom 2009. Laponia, Sweden, 2009.

[IPT12] Swati Ittan, Gaurav Paruthi, and William Thies. “Mapping Large Educational Websites to Interactive DVDs”. In: *Technology for Education (T4E), 2012 IEEE Fourth International Conference on*. IEEE. 2012, pp. 196–199.

[IT03] ITU-T. *One-way transmission time*. Recommendation G.114. Geneva: International Telecommunication Union, May 2003.

[IT07] ITU-T. *Definition of Quality of Experience*. Liaison Statement FG IPTV-IL-0050. accessed 2012-01-03. Geneva: International Telecommunication Union, 2007. URL: `http://www.itu.int/md/T05-FG.IPTV-IL-0050/en`.

[JBO03] Pedro Jordano, Jordi Bascompte, and Jens M Olesen. “Invariant properties in coevolutionary networks of plant–animal interactions”. In: *Ecology letters* 6.1 (2003), pp. 69–81.

[JFP04] Sushant Jain, Kevin Fall, and Rabin Patra. “Routing in a delay tolerant network”. In: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM ’04. Portland, Oregon, USA: ACM, 2004, pp. 145–158. ISBN: 1-58113-862-8.

[JLCCG07] Sewook Jung, Uichin Lee, Alexander Chang, Dae-Ki Cho, and Mario Gerla. “BlueTorrent: Cooperative Content Sharing for Bluetooth Users”. In: *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 47–56. ISBN: 0-7695-2787-6.

[Jou03] Yuh-Jzer Joung. “Quorum-Based Algorithms for Group Mutual Exclusion”. In: *IEEE Transactions on Parallel and Distributed Systems* 14 (2003), pp. 463–476.

[JOW+02a] P. Juang, H. Oki, Y. Wang, et al. “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet”. In: *Proc. of conference on Architectural support for programming languages and operating systems (ASPLOS)*. 2002.

[JT03] Mark Johnson and Peter Tyack. “A Digital Acoustic Recording Tag for Measuring the Response of Wild Marine Mammals to Sound”. In: *IEEE Journal of Oceanic Engineering* (2003).

[JTHL05a] Jehn-Ruey Jiang, Yu-Chee Tseng, Chih-Shun Hsu, and Ten-Hwang Lai. “Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks”. In: *Mob. Netw. Appl.* 10 (1-2 2005), pp. 169–181. ISSN: 1383-469X.

[Kal90] Maria-Theresia Kaltwasser. “Acoustic signaling in the black rat (Rattus rattus).” In: *Journal of Comparative Psychology* 104.3 (1990), p. 227.

[KCC05] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. “MANET simulation studies: the incredibles”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 9.4 (2005), pp. 50–61.

[KK00] B. Karp and HT Kung. “GPSR: greedy perimeter stateless routing for wireless networks”. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, pp. 243–254.

[KLR10] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan (Raj) Rajkumar. “U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol”. In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN ’10. Stockholm, Sweden: ACM, 2010, pp. 350–361. ISBN: 978-1-60558-988-6.

[KT10] Ravi Kumar and Andrew Tomkins. “A Characterization of Online Browsing Behavior”. In: *Proceedings of the 19th International Conference on World Wide Web*. WWW ’10. Raleigh, North Carolina, USA: ACM, 2010, pp. 561–570. ISBN: 978-1-60558-799-8.

[LCGZ05] Jason Lebrun, Chen nee Chuah, Dipak Ghosal, and Michael Zhang. “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks”. In: *In IEEE Vehicular Tech. Conf.* 2005, pp. 2289–2293.

[LD07] Anders Lindgren and Avri Doria. “Experiences from Deploying a Real-Life DTN System”. In: *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*. 2007, pp. 217 –221.

[LDS03] Anders Lindgren, Avri Doria, and Olov Schelén. “Probabilistic routing in intermittently connected networks”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 7 (3 2003), pp. 19–20. ISSN: 1559-1662.

[LDS04] Anders Lindgren, Avri Doria, and Olov Schelén. “Probabilistic Routing in Intermittently Connected Networks”. In: *Service Assurance with Partial and Intermittent Resources*. Vol. 3126/2004. LNCS. Springer, 2004, pp. 239–254.

[LFC05] J. Leguay, T. Friedman, and V. Conan. “DTN routing in a mobility pattern space”. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM. 2005, p. 283.

[LH09] Anders Lindgren and Pan Hui. “The quest for a killer app for opportunistic and delay tolerant networks”. In: *Proceedings of the 4th ACM workshop on Challenged networks*. ACM. 2009, pp. 59–66.

[Lin11] Anders Lindgren. “Social networking in a disconnected network: fbDTN: facebook over DTN”. In: *Proceedings of the 6th ACM workshop on Challenged networks*. CHANTS ’11. Las Vegas, Nevada, USA: ACM, 2011, pp. 69–70. ISBN: 978-1-4503-0870-0.

[LM98] SD Lang and LJ Mao. “A torus quorum protocol for distributed mutual exclusion”. In: *Proc. of the 10th Int’l Conf. on Parallel and Distributed Computing and Systems*. Citeseer. 1998, pp. 635–638.

[LSZM04] T. Liu, C.M. Sadler, P. Zhang, and M. Martonosi. “Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet”. In: *Proceedings of the 2nd international conference on Mobile systems, applications, and services.* ACM. 2004, pp. 256–269. ISBN: 1581137931.

[Mae85] Mamoru Maekawa. “A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems”. In: *ACM Trans. Comput. Syst.* 3 (2 1985), pp. 145–159. ISSN: 0734-2071.

[MB01] Michael J. McGlynn and Steven A. Borbash. “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks”. In: *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing.* MobiHoc ’01. ACM, 2001, pp. 137–145.

[MTK02] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. “Assessment of VoIP quality over Internet backbones”. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.* Vol. 1. 2002, 150 –159 vol.1.

[Now78] DA Nowitz. *Uucp implementation description.* Vol. 2. Bell Laboratories. In UNIX Programmer’s Manual, 1978.

[NS07] G. Nagy and S. Salhi. “Location-routing: Issues, models and methods”. In: *European Journal of Operational Research* 177.2 (2007), pp. 649–672.

[OK04a] Jörg Ott and Dirk Kutscher. “Drive-thru Internet: IEEE 802.11b for "automobile" users”. In: *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies.* Vol. 1. 2004, 4 vol. (xxxv+2866).

[OK04b] Jörg Ott and Dirk Kutscher. “Why Seamless? Towards Exploiting WLAN-based Intermittent Connectivity on the Road”. In: *Proceedings of TERENA Networking Conference, TNC 2004.* 2004.

[PTR09] J.M. Pujol, A.L. Toledo, and P. Rodriguez. “Fair routing in delay tolerant networks”. In: *INFOCOM 2009, IEEE.* IEEE. 2009, pp. 837–845.

[RBK07] Ram Ramanathan, Prithwish Basu, and Rajesh Krishnan. “Towards a formalism for routing in challenged networks”. In: *CHANTS ’07: Proceedings of the second ACM workshop on Challenged networks.* Montreal, Quebec, Canada: ACM Press, 2007.

[RFC2397] L. Masinter. *The data URL scheme.* RFC 2397. Internet Engineering Task Force, Aug. 1998. URL: `http://www.rfc-editor.org/rfc/rfc2397.txt`.

[RFC4737] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser. *Packet Reordering Metrics.* RFC. RFC Editor, 2006. URL: `https://tools.ietf.org/html/rfc4737`.

[RFC5050] K. Scott and S. Burleigh. *Bundle Protocol Specification*. RFC 5050. Internet Engineering Task Force, Nov. 2007. URL: http://www.rfc-editor.org/rfc/rfc5050.txt.

[RFC7540] M. Belshe, R. Peon, and M. Thomson. *Hypertext Transfer Protocol Version 2 (HTTP/2)*. RFC. RFC Editor, 2015. URL: http://httpwg.org/specs/rfc7540.html.

[Sin38a] James Singer. “A theorem in finite projective geometry and some applications to number theory”. In: *Trans. Amer. Math. Soc.* 43 (1938), pp. 377–385.

[SMPW11] Sebastian Schildt, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. “IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation”. In: *Electronic Communications of the EASST* 37 (2011).

[SO11] Immanuel Scholz and OpenStreetMap community. *Java OpenStreetMap Editor*. 2011. URL: http://josm.openstreetmap.de/.

[Sor09] J.M. Sorber. “Into the wild: taming uncertainty in perpetual mobile networks”. In: *Proceedings of the 2009 MobiHoc S 3 workshop on MobiHoc S 3*. ACM. 2009, pp. 41–44.

[SPMC04] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler. “Lessons from a Sensor Network Expedition”. In: *Proc. of European Workshop on Wireless Sensor Networks*. 2004.

[SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. “Spray and wait: an efficient routing scheme for intermittently connected mobile networks”. In: *WDTN ’05*. New York, NY, USA: ACM Press, 2005.

[SQBB10] C. Song, Z. Qu, N. Blumm, and A.L. Barabasi. “Limits of predictability in human mobility”. In: *Science* 327.5968 (2010), p. 1018.

[SZ89] K. Stephenson and M. Zelen. “Rethinking centrality: Methods and examples”. In: *Social Networks* 11 (1989).

[TTLW97] Wai-Shing Luk Tien-Tsin, Wai shing Luk, and Tien tsin Wong. “Two New Quorum Based Algorithms for Distributed Mutual Exclusion”. In: *17th International Conference on Distributed Computing Systems*. IEEE, 1997, pp. 100–106.

[VIF06] A. Vlavianos, M. Iliofotou, and M. Faloutsos. “BiToS: Enhancing BitTorrent for Supporting Streaming Applications”. In: *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. 2006, pp. 1–6.

[Voi97] H-M Voipio. “How do rats react to sound?” In: *Scandinavian Journal of Laboratory Animal Science.* Supplement 24.1 (1997), pp. 1–80.

[WF74] Robert A. Wagner and Michael J. Fischer. “The String-to-String Correction Problem”. In: *J. ACM* 21 (1 1974), pp. 168–173. ISSN: 0004-5411.

[WHFP09] L. Wood, P. Holliday, D. Floreani, and I. Psaras. "Moving data in DTNs with HTTP and MIME". In: *2009 International Conference on Ultra Modern Telecommunications Workshops*. 2009, pp. 1–4.

[WPPSM07] L. Wood, C. Peoples, G. Parr, B. Scotney, and A. Moore. "TCP's protocol radius: the distance where timers prevent communication". In: *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*. 2007, pp. 163–167.

[Zha06] Zhensheng Zhang. "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges". In: *Communications Surveys Tutorials, IEEE* 8.1 (2006), pp. 24 –37. ISSN: 1553-877X.

[Zha12] Jie Zhang. *Tutorial on Small Cell/HetNet Deployment, Part 1: Evolutions towards small cell and HetNet*. In: Globecom'12 Industry Forum. 2012.

Online Resources

[Ale16] Alexa Internet Inc. *Alexa Top 500 Global Sites*. 2016. URL: `http://www.alexa.com/topsites` (visited on 04/25/2016).

[And16] Android Open Source project. *Building for Billions*. 2016. URL: `https://developer.android.com/distribute/essentials/quality/billions.html#network-offline`.

[App16] Apple Inc. *About Networking*. 2016. URL: `https://developer.apple.com/library/prerelease/content/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/`.

[Blu10] Bluetooth SIG. *Bluetooth Technical Core Specification Version 4.0*. 2010. URL: `http://www.bluetooth.org/Technical/Specifications/adopted.htm`.

[Cla03] Clay Shirky. *Permanet, Nearlynet, and Wireless Data*. 2003. URL: `http://shirky.com/writings/permanet.html`.

[Esc16] Escape Artists, Inc. *Conditions of Use: Escape Pod*. 2016. URL: `http://escapepod.org/conditions-of-use/` (visited on 04/28/2016).

[Har05] Harvard University. *Motelab*. 2005. URL: `http://motelab.eecs.harvard.edu/`.

[Ily16a] Ilya Grigorik. *Preload*. Mar. 2016. URL: `https://www.w3.org/TR/2016/WD-preload-20160321/`.

[Ily16b] Ilya Grigorik. *Resource Hints*. Feb. 2016. URL: `https://www.w3.org/TR/2016/WD-resource-hints-20160225/`.

[Jam06] James Scott, Richard Gass, Jon Crowcroft, Pan Hui, Christophe Diot, and Augustin Chaintreau. *CRAWDAD data set cambridge/haggle (v. 2006-09-15)*. 2006. URL: `http://crawdad.cs.dartmouth.edu/cambridge/haggle`.

[Mar15] Marc Blanchet. *DTN Use Cases*. 2015. URL: http://trac.tools.ietf.org/wg/dtn/trac/wiki/DtnUseCases (visited on 02/15/2015).

[Nat05a] Nathan Eagle and Alex S. Pentland. *CRAWDAD data set mit/reality (v. 2005-07-01)*. 2005. URL: http://crawdad.cs.dartmouth.edu/mit/reality.

[Ope11a] OpenStreetMap community. *Indoor Mapping – OpenStreetMap Wiki*. 2011. URL: http://wiki.openstreetmap.org/wiki/Indoor_Mapping.

[Ope11b] OpenStreetMap community. *OpenStreetMap, The Free Wiki World Map*. 2011. URL: http://www.openstreetmap.org/.

[Pil10] Pillip Levis et al. *TinyOS*. 2010. URL: http://www.tinyos.net/.

[Wik16] Wikimedia! The Wikimedia Foundation Inc. *List of Wikipedias*. 2016. URL: https://meta.wikimedia.org/wiki/List_of_Wikipedias (visited on 04/25/2016).

[Win16] Windows Apps Team. *Offline Apps: what to do when your user disconnects*. 2016. URL: https://blogs.windows.com/buildingapps/2016/07/25/offline-apps-what-to-do-when-your-user-disconnects/.