Diss. ETH No. 16681

Field-based Routing and its Application to Wireless Ad Hoc Networks

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Science

presented by

Vincent L. C. Lenders

Dipl. El.-Ing. ETH
born October 19, 1977
citizen of Belgium

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Jim Kurose, co-examiner
Prof. Marco Conti, co-examiner

2006

Abstract

Wireless ad hoc networks consist of autonomous, and possibly mobile, devices communicating over wireless links without any support from a fixed infrastructure. This recently proposed networking paradigm allows for completely new types of applications, particularly in areas where no networking infrastructure is available. However, such networks are greatly challenged by the possible movement of the participating nodes and the highly variable wireless channel conditions, making the task of packet routing particularly difficult to achieve.

In this thesis, we therefore develop a new concept for robust routing in wireless ad hoc networks. Inspired by the field theory from physics, we model data packets as test charges diffusing along the steepest gradient of a potential field created by the destination nodes. This approach is very efficient in networks with frequent topology changes and guarantees loop-freedom. We call our concept *field-based routing* and use it in different contexts/-contributions of the thesis.

The first contribution is the proposal and evaluation of density-based anycast routing, a novel anycast routing scheme which takes into account the density of the group members in the routing direction. We use field-based routing to model and compare density-based routing versus traditional shortest-path anycast routing, which solely considers the distance to the closest group member for routing. We show that density-based routing outper-

forms shortest-path anycast routing when the nodes are mobile or when the wireless channel conditions are highly variable.

The second contribution is a service discovery protocol for wireless ad hoc networks. Again, we use field-based routing and model service instances as positive point charges and service queries as negative test charges. A query is therefore routed along the steepest gradient of the field to a matching service instance. The advantage of this approach lies in the combination of routing and service discovery in a single protocol, hence reducing the protocol overhead. Furthermore, field-based service discovery allows for service differentiation, allowing to perform load balancing or more generally, allowing clients to select the "best" service according to their location.

The last contribution is the proposal of a new network architecture for wireless ad hoc networks. Our architecture does not enforce any binding between clients and services such as in IP networks. The problem with enforcing a client-service binding is that it prevents clients from switching to a better service alternative as the nodes, and possibly also the services, are moving. Rather, we provide three binding semantics in our architecture that allow clients to decide whether a binding is required depending on the application context.

Kurzfassung

Drahtlose ad hoc Netze bestehen aus autonomen, oft mobilen Geräten, die ohne feste Infrastruktur miteinander kommunizieren. Solche Netzwerke ermöglichen die Entwicklung von völlig neuartigen Anwendungen, speziell an Orten, an denen keine Netzwerk-Infrastruktur vorhanden ist. Allerdings stellt dieser neue Netwerktyp eine besondere Herausforderung für die Kommunikation zwischen den Endgeräten dar, da alle Knoten mobil sein können und die Qualität des drahtlosen Kommunikationsmediums stark variieren kann. Diese Dynamik erschwert besonders die effiziente Implementierung der Leitweglenkung (Routing).

In dieser Dissertation entwickeln wir ein neues Verfahren für robustes Routing in ad hoc Netzen. Inspiriert von der Feldtheorie in der Physik, modellieren wir Datenpakete als Probeladungen, die in einem Potentialfeld entlang des steilsten Gradienten diffundieren, welches durch die Zielknoten aufgebaut wird. Wir nennen dieses neue Konzept *Field-based Routing* und verwenden es in den verschiedenen Beiträgen dieser Dissertation.

Der erste Beitrag ist der Vorschlag und die Evaluierung eines dichte-basiertem Anycast-Routing Ansatzes. Die Idee des Anycast-Routings besteht darin, Pakete in Abhängigkeit der Dichte der Gruppenteilnehmern zu leiten. Anhand des Modells für Field-based Routing sind wir in der Lage, dichte-basiertes Anycast-Routing zu modellieren und mit traditionellen shortest-path Anycast-Routing Algorithmen zu vergleichen. Wir zeigen, dass der

dichte-basierte Ansatz zuverlässiger arbeitet als shortest-path Anycast-Routing, wenn die Knoten mobil sind oder wenn die Verbindungsqualität zwischen den Geräten stark variiert.

Der zweite Beitrag dieser Arbeit ist ein neues Protokoll, welches ermöglicht, Dienste in einem drahtlosen ad hoc Netzwerk zu finden. Wieder benützen wir Field-based Routing und modellieren Dienste als positive Punktladungen und Dienstanfragen als negative Probeladungen. Durch den felder-basierten Ansatz werden Dienstanfragen entlang des steilsten Anstieges zu einem passenden Dienst geleitet. Durch die geschickte Kombination von Routing und Dienstsuche werden individuelle Nachrichten für Routing und Dienstsuche überflüssig. Zudem kann felder-basierte Dienstsuche Dienste nach ihrer Kapazität oder ihrer momentanen Belastung unterscheiden, was einem erlaubt, Anfragen zu leistungsfähigeren oder weniger stark benützten Diensten zu leiten.

Als letzten Beitrag liefern wir einen Vorschlag für ein neues Kommunikationsparadigma für drahtlose ad hoc Netze. Unsere vorgeschlagene Architektur erlaubt es, im Gegesatz zu bestehenden IP Netwerken, auf die Bindung der Klienten an einen bestimmten Dienst zu verzichten. Diese Bindung stellt deshalb ein Problem dar, weil dadurch Klienten davon abgehalten werden, bessere Dienste zu verwenden, wenn sich Klienten oder Dienste bewegen. Stattdessen enwickeln wir einen Ansatz, der verschiedene Bindungssemantiken anbietet. Diese ermöglichen dem Klienten, die Art der Bindung in Abhängigkeit des Applikationskontexts zu wählen.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There has been considerable growth within the last years in the number of small devices such as cell phones, PDAs, laptops, small sensor nodes, music players, and game consoles. The variety of their incorporated *short-range wireless* networking capabilities allows us to envisage new applications. We can imagine that *ad hoc* and *autonomous* networks spring up around the movement of people in buildings, campus, public transportation vehicles, corporate environments, at conferences, or even whole cities. Contextual applications, location-based services, or basic applications like content distribution could take advantage of such an ad hoc network.

While the massive deployment of regulated and fixed wireless infrastructures such as GSM, EDGE or UMTS networks already have made some of those applications reality today, there is still a need for more flexible and ad hoc networks which do not rely on fixed base stations for communication between the mobile nodes. Our visionary network is an autonomic network of participating nodes which cooperate to provide the basic communication prim-

itives including for example multihop routing or service discovery without the need of any support from a fixed infrastructure. The motivations for ad hoc networks are manifold:

- *Cost:* The deployment cost of a fixed wireless infrastructure is expensive and often reflected in the high communication costs for the end users. An ad hoc network could be used as an alternative for "low-cost" communication.

- *Disaster or hostile environments:* An ad hoc network could be the only communication means in hostile environments, or after a natural disaster such as for example an earthquake.

- *Energy:* An ad hoc network could reduce the energy consumption of the mobile nodes (which is scarce because nodes are mostly battery powered) by exploiting local communication opportunities instead of establishing wireless connections over large distances.

- *Regulation:* Regulated networks are often controlled by the state. Ad hoc networks operating over unlicensed frequency bands could alleviate this problem and allow people to communicate in a way which prevents controlling from a third party.

- *Poor or discriminated regions:* The provisioning of fixed wireless services in dense urban areas is very ubiquitous. However, users which reside in less populated areas or in regions with a low income are often discriminated.

Before describing specific research problems and the contributions of this thesis, we first describe four specific and promising application scenarios for wireless ad hoc networks that could profit from the results of this thesis.

1.2 Applications

In this thesis, we focus on applications for wireless ad hoc networks that aim at collecting data, or provide access to services. As we will see later, the communication pattern for these type of applications is mainly *anycast* from a sender node (a client or a data source) to *any* receiver node that exhibits certain properties (a service or a data sink). In contrast, traditional networks such as the telephony system or the Internet, have been designed to support point-to-point (or *unicast*) connectivity. This indeed is the communication pattern of the most common application of these networks, namely voice communication and web server access.

1.2.1 Service Access

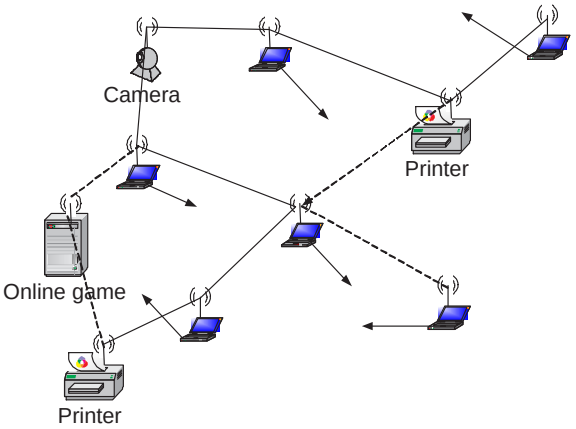


Figure 1.1: *Service access network.*

Wireless ad hoc networks could be used to support nomadic users with information or services as they move. In this application (see Figure 1.1), we envision a network including different

types of services such as for example printers, cameras, games, or information sources that people access with their mobile devices. Note that the specific server in the network that provides a service is often irrelevant to the user, but the utility of the service is directly related to its proximity.

1.2.2 Sensor Networks

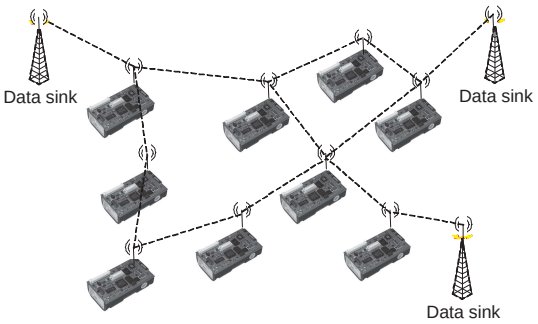


Figure 1.2: *Sensor network.*

A sensor network is an ad hoc network of small wireless sensor devices which communicate over radio links (see Figure 1.2). Such networks have been proposed for environmental monitoring [**?**], wildlife habitat monitoring [**?**], building automation or military contexts. In any context, each node senses its environment and reports its data to specific nodes called data sinks. These data sinks are typically connected to a fixed network and can possibly be accessed over the Internet. Routing in a sensor network with multiple data sinks is the task of delivering sensor data to any data sink (anycast).

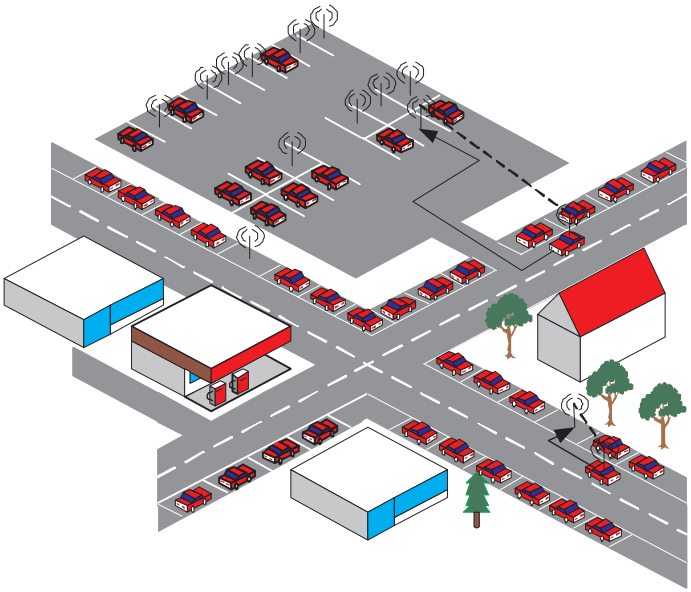


Figure 1.3: *Vehicular network.*

1.2.3 Vehicular Networks

Vehicular networks have received a lot attention recently. The idea of networking automobiles together is to support the driver with useful information that he is not able to obtain easily by himself. Typical information of that kind could be information about traffic jams, about weather conditions, or car accidents. In Figure 1.3, a parking assistance application is shown. Parking spaces, as well as cars, are equipped with wireless devices and organize each other to guide drivers to any free space (e.g., the closest one).

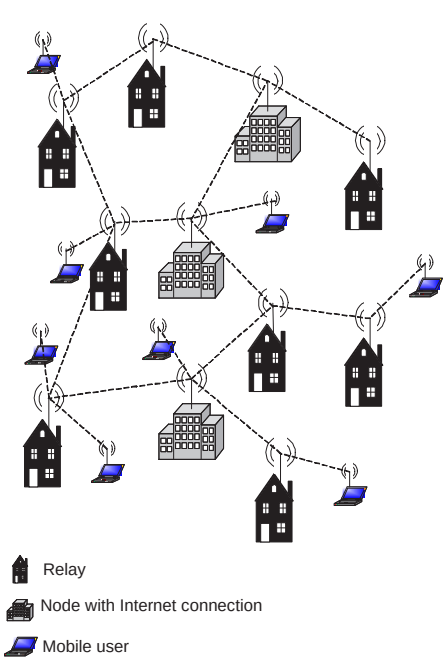


Figure 1.4: *Mesh network.*

1.2.4 Mesh Networks

A mesh network is a multihop wireless network that provides Internet connectivity to mobile users. Those type of networks have been deployed on top of house roofs in several cities (e.g., Berlin [**?**], Seattle [**?**], New York [**?**]) and research projects (e.g., at MIT [**?**]). Figure 1.4 shows an example. A few nodes (here pictured as buildings) have a connection to the Internet over a fixed infrastructure (typically DSL lines) and the remaining nodes (houses) act as relays. Mobile users (shown as laptops) use this infrastructure to connect to the Internet via the nodes with Internet connectivity. Anycast routing is a natural fit to route packets in a mesh network since the packets can be sent to the Internet

via any gateway node.

1.3 Research Problems

Wireless ad hoc networks are challenging in many ways. On the one hand, the wireless links and the mobility of the nodes introduce a high degree of dynamism in the network. On the other hand, the lack of a fixed infrastructure makes any centralized solutions inapplicable. We therefore cannot rely on the presence of dedicated central servers.

In this thesis, we tackle the following research problems with regard to wireless ad hoc networks:

- *How to apply concepts from field theory to routing in wireless ad hoc networks?* Our idea is to route data packets in the network in analogy to how a test charge diffuses in an electrical field. The main problem that arises is that fields in physics are continuous whereas a field in a network is discrete as it can only have values at individual nodes. We study the effects of this discretization.

- *How to perform anycast routing in the presence of mobile nodes and/or unreliable links?* This question is twofold. The first issue is to find the best routing strategy. The second issue is to design a technique and a control protocol that implements the desired routing strategy.

- *How to implement service discovery in an ad hoc network?* Mechanisms are required to discover services instances in a fully distributed way. When there are many service instances providing the same service, how can we discover a close service with a high capacity in an efficient way, and perform load balancing?

- *How should a network architecture for ad hoc networks be designed to support mobility?* A flexible and autonomic network architecture is needed to support different types of binding semantics between clients and services depending on the application requirements.

1.4 Contributions

The main contribution of our work is the design of field-based routing, a model for robust and loop-free routing, and its application to wireless ad hoc networks. The detailed contributions are listed below. We also list the conference or the journal where each contribution was published.

- We propose a robust and loop-free routing model based on the concept of potential fields [**?**]. Using the model, it is possible to deliver packets successfully to their destination, even when links break or nodes along the path to the destination move. We prove that our model is loop-free, and derive worst-case bounds for which we can guarantee that routing converges. We further show that in random wireless networks, the probability is very high that routing converges.

- We propose a new anycast routing strategy: density-based anycast [**?**]. Density-based routing considers the number of group members involved in the routing decision, in contrast to existing proximity-based routing techniques, which only consider distance information.

- We conduct a quantitative comparison of density- versus proximity-based routing [?]. We show that density-based anycast routing is more robust than proximity-based routing in a mesh network scenario and that in a highly dynamic sensor network, it finds even shorter paths.

- We design and evaluate a service discovery protocol for wireless ad hoc networks based on the field-based routing model [?, **?**]. The protocol combines routing and service discovery which reduces overhead compared to having two separate protocols. Furthermore, field-based service discovery can differentiate services based on their capacity or their current load, allowing to distribute the requests to "good" services or services with a low load.

- We propose a new networking architecture for ad hoc networks that relies on an anycast routing substrate [**?**]. We show the benefits of our approach and present a proof-of-concept implementation.

We have patented [**?**] the application of field-based routing for discovering services on dynamic network. Different industry partners have indicated their interest in the patent.

1.5 Outline

This thesis is structured as follows:

- **Chapter 2** reviews related work and highlights the novel aspects of this thesis compared to previous work.

- **Chapter 3** introduces our field-based routing approach. Detailed examples highlight the effect of individual parameters, and the approach is analyzed analytically.

- **Chapter 4** introduces density-based anycast routing and how we employ field-based routing to model anycast routing strategies. We compare density-based routing with proximity-based routing for mesh and sensor networks.

- **Chapter 5** describes our service discovery protocol based on field-based routing. Our implementation in the network

simulator GloMoSim, and the simulation results are also discussed.

- **Chapter 6** presents the overall communication architecture we propose on top of field-based routing and discusses the benefits and limitations compared to the TCP/IP protocol suite.
- **Chapter 7** concludes this thesis and provides details about two new research projects that plan to employ field-based routing as routing substrate.

Chapter 2

Related Work

In this chapter, we review related work that has influenced our work on field-based routing. We start by describing work on unicast routing in ad hoc networks. Then, we present approaches that use the concept of fields or gradients for routing. Finally, we survey related work on anycast, on service discovery, and on autonomic network architectures for mobile networks.

2.1 Unicast Routing Protocols for Ad Hoc Networks

Enormous research efforts have been put in the design and optimization of unicast routing protocols for ad hoc networks. We survey the most common protocols in the following.

Proactive routing protocols establish routing state in the network before the routes are required. Routing protocols using this approach are DSDV [**?**], OSLR [**?**], and TBRPF [**?**]. On the other hand, reactive routing protocols establish routes on demand only when they are needed. DYMO [**?**], AODV [**?**] and DSR [**?**] all employ such a scheme. Hybrid approaches such as ZRP [**?**] that combine the benefits of both approaches have also been proposed.

Another kind of routing technique is geographical routing. Geographical routing uses geographical information (e.g., coordinates obtained from the global positioning system (GPS)) to forward packets to their destination. The Location-Aided Routing (LAR) protocol [**?**] belongs to this category of routing protocols.

The focus of our work is not on unicast routing but on anycast routing.

2.2 Field- and Gradient-based Routing

Most distributed routing protocols are based on two basic techniques: distance vector routing or link state routing. Distance vector based routing protocols such as for example BGP [**?**], RIP [**?**], DSDV [?], or AODV [?] to name just a few, establish routing state by exchanging distance information to destinations between neighbors. On the other hand, link state based routing protocols (e.g., OSPF [**?**], OSLR [?]) exchange information about the complete set of links that exist in the network. Each node has a complete view of the network and computes the shortest path locally based on for example Dijkstra shortest path algorithm. These two routing techniques are well studied in the literature and employed in different kinds of existing networks (e.g., the Internet).

In this thesis, we consider a different routing technique based on the concept of fields. Field-based routing or gradient-based routing is far less explored as the abovementioned techniques. In the following, we present other works which used fields or gradients for routing in data networks, and explain how these works differ from ours.

2.2.1 TORA

Vincent Park and Scott Corson introduced the Temporally Ordered Routing Algorithm (TORA) [**?**] in 1997 for routing in mobile ad hoc networks. TORA establishes a directed acyclic graph (DAG) rooted at the destination node which can be viewed as a potential field. The DAG is constructed by assigning a "height" (or a potential value as we call it in our work) to each node and then determining the steepest ascent by looking at the relative height between neighboring nodes. The protocol tries to minimize routing overhead, and for this purpose, applies a technique called *link reversal* to repair broken links in the DAG as nodes move. Link reversal consists of reversing individual links in the DAG until it is routed at the destination again.

TORA was designed for unicast routing and operates only on DAGs (or potential fields) with a single destination node. The authors do not show how to extend the model for multiple destinations and it remains unclear if the proposed protocol as such can be extended for this purpose.

2.2.2 AntNet

AntNet is a routing algorithm inspired from the behavior of ants proposed by Gianni Di Caro and Marco Dorigo in 1997 [**?**]. The algorithm mimics how real ants find shortest paths by following a pheromone trail (or gradient) deposited by other ants. At the beginning, when no pheromone trails are available, the ants move randomly. The algorithm starts to converge as soon as one ant has discovered a path to the desired destination.

Ant-based routing belongs to the class of statistical routing algorithm since it introduces randomness to find routes. In our work, fields and gradients are established in a deterministic way. As such, AntNet was proposed for unicast routing but we believe that it could be easily adapted for anycast routing. However, it

remains unclear if AntNet would then be able to achieve density-based routing strategies as our field-based routing model does.

2.2.3 Directed Diffusion

Directed Diffusion [**?**] was proposed in 2000 by Chalermek Intanagonwiwat et. al. as a communication paradigm for sensor networks. Sensor data is named and "diffuses" according to pre-established interest gradients towards the sinks (nodes that are interested in the data). One feature of directed diffusion is the ability to aggregate information according to the interests of the nodes.

Directed Diffusion supports multiple destinations but the goal of the system is to deliver the data to all of the interested nodes (multicast). In our work, we focus on delivering data to the best destination node (anycast). Although the work inspired us and many researchers in the area of sensor networks, the work does not provide a mathematical model as we do in our thesis.

2.2.4 Routing using Potentials

In 2003 [**?**], Anindya Basu et. al. used potential based routing to design a dynamic, traffic-aware routing algorithm for the Internet. The basic idea is to avoid congested areas in the network by modelling the queue length of routers as an input for the potential values of the potential field. In their case, steepest gradient routing results in routing through routers which are less congested.

The authors give a complete model to describe and evaluate potential-based routing. However, they work focusses on unicast routing in fixed wired networks, whereas our in our work, we use potential fields for anycast routing in dynamic, wireless, and mobile networks.

2.2.5 Packetostatics

The approach presented in [**?**] relies on the electrostatic field to route flows in sensor networks. They show that the optimal distribution of nodes induces a traffic flow identical to the electrostatic field that would exist if the sources and sinks of traffic were substituted with an appropriate distribution of electric charge. The authors of this work assume a massively dense sensor network which justifies their analysis using the continuous electrostatic field. In our work, we do not make this strong assumption, and consider the discrete case. Furthermore, their work is focussed on finding the optimal node placement given a set of flows, where our work is really about finding optimal routes given a network topology.

2.3 Anycast Routing Protocols

Anycast is a delivery mode whereby a packet is sent to one of a group of hosts. In this section, we discuss research efforts around anycast routing for the Internet and for mobile networks.

2.3.1 In the Internet

IP anycast was proposed as an RFC [**?**] for the Internet in 1993 to simplify the task of finding an appropriate server when several servers support the desired service. They proposed to assign an unicast IP address to multiple hosts, and advertising it into the routing infrastructure from all the hosts which share the same address. Later, it was incorporated into the IPv6 addressing architecture [**?**].

The reasons why IP anycast is not widely deployed nowadays in the Internet is mainly due to two problems. IP anycast, as originally proposed, does not scale well and is hard to deploy in the core of the Internet at a large scale. As an attempt, Katabi

et. al. proposed GIA [**?**] as a scalable IP architecture. However, this approach requires a change in the core Internet routers and puts a severe dent on the practical appeal of the approach. For this reason, groups of researchers proposed application-layer solutions. The work in [**?**], i3 [**?**], and PIAS [**?**] were for example proposed as application-layer solutions.

Although scalability and ease of deployment are fundamental to the success of anycast in the Internet, it is less important for infrastructure-less or self-organized networks such as mobile, ad hoc networks, or sensor networks which is the focus of this thesis. We expect these kind of networks to be smaller (hundred to thousands of nodes compared to millions of nodes in the Internet) and to be used for "local" communication (e.g. for communication within a city or a group of persons). Furthermore, self-organizing networks often do not rely on an existing routing infrastructure, which makes the problem of protocol deployment much easier.

2.3.2 In Mobile Networks

Vincent Park et. al. described in [**?**] and [**?**] how to extend known unicast routing techniques such as link state and distance vector routing for anycast delivery. Jianxin Wang et al. proposed in [**?**] and [**?**] to extend AODV [?] and DSR [?] respectively, to support anycast delivery in mobile ad hoc networks.

These protocols all rely on unicast routing techniques, and as a result, they implement the same routing strategy: routing to the closest group member over the shortest path. In our thesis, we introduce and explore a new family of anycast routing strategies: proximity-based routing, density-based routing, and combinations of both.

2.4 Service Discovery

We differentiate service discovery solutions that operate at the application-layer and those that combine the routing task (network-layer).

2.4.1 Application-Layer

Several industrial standards for service discovery emerged at the end of the nineties. Sun's Jini [**?**] provides a complete framework for spontaneous distributed computing. Services are defined as Java interfaces and subscribe to directories (lookup servers) before clients download a service proxy and access these services over Java's Remote Method Invocation (RMI). Universal Plug and Play (UPnP) [**?**] as pushed by Microsoft provides similar functions but is based on XML instead of Java interfaces. The IETF with its Service Location Protocol (SLP) [**?**] has also put some effort towards standardizing a service discovery framework for the Internet. At around the same time, researchers from Berkeley developed an architecture for secure service discovery [**?**] based on XML. All these systems were designed for infrastructure based networks such as the Internet. As such, these schemes will not work in mobile ad hoc networks.

Konark [**?**] was designed to support service discovery in wireless ad hoc networks. Services are expressed using XML. The service delivery itself is based on SOAP. As Konark is an application-layer approach, it assumes an underlying routing substrate.

2.4.2 Network-Layer

The Intentional Naming System (INS) [**?**] was proposed in 1999 by William Adjie-Winoto et. al. from MIT. In that approach, service requests by clients are directly routed towards a matching service instance without an intermediate lookup to discover its

address. To route these requests, a resolver network of dedicated INS resolvers is proposed. The novelty of this work is more on the design of the naming architecture whereas our thesis focuses on the routing of service queries and the selection of a specific service instance. Furthermore, INS was designed for wired networks, whereas our approach makes usage of the broadcast nature of the wireless channel.

The idea of combining service discovery with MANET routing protocols was formulated in 2002 as an RFC [**?**] by Koodli and Perkins. The basic idea is to add service information in route request messages from on-demand ad hoc routing protocols such as AODV [?] or DSR [?]. A drawback from this approach is that each service request generates a flooding in the network.

Kozat and Tassiulas [**?**] proposed one year later a service discovery mechanism targeted at mobile ad hoc networks. A virtual backbone is constructed dynamically, assuring that all nodes are part, or are at least one hop away from this backbone. The proposed service discovery system does not provide mechanisms for service selection or discrimination when multiple service instances of the same type coexist.

The Bluetooth specification [**?**] defines a service discovery protocol (SDP) to discover the services a device offers. The protocol is based on a simple query-reply mechanism. As it is defined today, SDP works only over one hop. Furthermore, the protocol is specific to the Bluetooth technology.

2.5 Network Architectures

The literature abounds with proposals for architectural fixes of IP. These offer solutions to a number of problems such as mobility [**?**, **?**], routing [**?**, **?**], naming/addressing/layering [**?**, **?**, **?**, **?**, **?**], binding [?], security [**?**, **?**], or quality of service [**?**, **?**, **?**]. However, all the above proposals choose of improving current ISP infras-

tructures. Our aim is to design a new architecture for ad hoc networks.

Many architectural problems of IP can be solved at the application layer [?] as overlays. However, these solutions suffer from efficiency problems when applied to wireless ad hoc networks as they introduce a significant stretch of the path lengths. A more efficient solution which we also propose in this thesis is to make use of a cross-layer design. For example the works in [**?**, **?**] use information from the application layer at the routing layer to avoid the path length stretch from strictly layered architectures.

To cope with the diversity of the different semantics and addressing schemes of possible sensor network applications, the authors of [**?**, **?**] proposed a sensor network architecture that lowers the common communication abstraction down to single-hop communication instead of multi-hop as with IP or our architecture.

Publish-subscribe systems have been proposed to decouple the process of sending with the process of receiving in networks. In a publish-subscribe system (e.g., TIB / RENDEZVOUS [**?**]), processes can subscribe to messages containing information on specific subjects, while other processes produce (i.e. publish) such messages. For this reason, field-based routing can be viewed as a publish-subscribe system. The clients would publish requests while the service providers subscribe to those by establishing a potential field. The publish-subscribe systems so far have been researched and developed mostly in fixed networks. Our approach takes full advantage of the broadcast nature of wireless radio where traditional publish-subscribe systems are mostly built as overlays over IP.

Silvia Ratnasamy, Scott Shenker, and Steven McCanne suggest in [**?**] that architectural support for global anycast is a missing part to IP. They claim that by the deploying anycast, the Internet architecture would be evolvable.

Chapter 3

Field-based Routing

This chapter presents the concept of field-based routing. We first describe the model and show why it is robust to link and node failures. Then, we use simple examples to discuss how the model parameters affect routing decisions. At the end of the chapter, we analyze field-based routing analytically to prove loop-freeness and assess convergence properties.

3.1 Model Description

3.1.1 Overview

The routing model is inspired from potential fields in physics (e.g., the gravitation or the electric potential). In physics, a potential field describes the potential energy associated with each point in space. By analogy, we use a potential field to model the "energy" of every node. We achieve this by creating a scalar field around each destination node which decreases relative to its distance. Data packets are routed along the steepest gradient of the destination's field, corresponding the path with the lowest required energy. The steepest gradient can easily be determined lo-

cally by comparing the potential value of the neighboring nodes. The direction of the steepest gradient is towards the neighbor with the highest potential value.

Fields are not restricted to one specific destination node. Different nodes can contribute to the same potential field. In that case, the potential field is obtained in the same way as in physics by creating the linear superposition of the influence from the individual nodes of the group. Packets following the steepest gradient will then be delivered to any node of this group.

3.1.2 Potential Fields

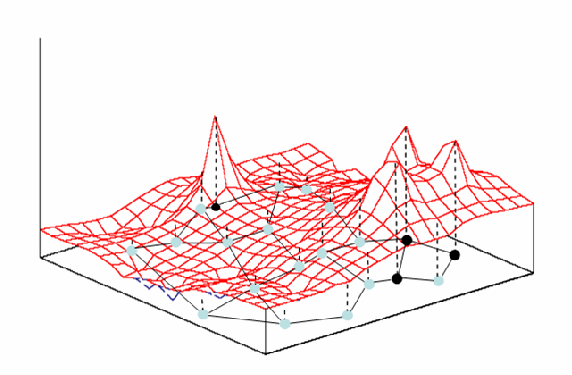


Figure 3.1: *Example potential field. Black nodes represent entities. Grey nodes are relay nodes.*

In physics, the electric potential at position $\vec{r}$ which relates to a point charge Q_j located at $\vec{r_j}$ is $\varphi_j(\vec{r}) = \frac{1}{4\pi\varepsilon}\frac{Q_j}{|\vec{r}-\vec{r_j}|}$. In analogy to this definition, we define the potential field of an entity[1]. If Q_j

[1]We use in this chapter the quite general term "entity" to describe a node independent of its application. An entity is referred to as a group member when discussing anycast routing in Chapter 4, and to a service instance in Chapter 5.

is the capacity of an entity at node n_j (the capacity of an entity could for example model the current availability of a service or its performance), then the potential value at an arbitrary node n in the network is defined as

$$\varphi_j(n) = \frac{1}{c} \cdot \frac{Q_j}{d_j(n)} \tag{3.1}$$

where $d_j(n)$ is the network distance between n and n_j, and c is a constant value. The network distance could be any metric as long as the distance function is strictly monotonously decreasing. In our work however, we only use the number of hops as the distance metric. Other possible metrics are typically the transmission delay, or a metric capturing the load or congestion in the network.

The main difference between the electric potential field and our definition is that the electric field is continuous whereas our definition of the potential is discrete (only defined at the network nodes). We will see later in this chapter that the discretization of the field can cause undesired local maxima.

For not restricting ourselves to a specific potential function such as the electric potential, we define the potential function more generally as

$$\varphi_j(n) = \frac{Q_j}{d_j^k(n)} \qquad ,0 < k < \infty \tag{3.2}$$

where k is a positive real number, and c is one, as the absolute value of the potential is not of interest. Note that the exponent k determines how quickly the potential decreases over distance. For high values of k, the field is steep whereas it becomes flatter for small values of k. We will analyze the effect of k on the routing in Chapter 4.

Different entities of a same group contribute to the same potential field. To stick to the physical analogy, we define the poten-

tial field of a group $\mathcal{N}$ as the linear superposition of the potential fields of all group entities:

$$\varphi(n) = \sum_{j \in \mathcal{N}} \varphi_j(n) = \sum_{j \in \mathcal{N}} \frac{Q_j}{d_j^k(n)} \tag{3.3}$$

With this definition, the potential field distribution resembles to a landscape with peaks ($\varphi \to \infty$) at every entity since $d_j(j) = 0, \forall j \in \mathcal{N}$. An example is drawn in Figure 3.1 for a group of four entities. The black nodes represent entities and the grey nodes are relays.

3.1.3 Gradient-based Routing

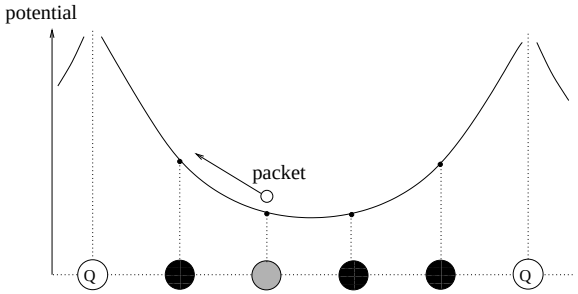


Figure 3.2: *Routing along the steepest gradient.*

The previously defined potential field can be used to route packets to any entity in the network. The routing mechanism is similar to field diffusion in physics. With field diffusion, an element (e.g., a test charge in a electrical field) is always attracted towards the steepest gradient of the field. If the element is free to move, it will diffuse along the steepest gradient until it arrives at a field maximum. In the same manner, we route packets along the steepest gradient of the potential field. Figure 3.2 illustrates how this works. The potential field resulting from the two entities

with capacity Q is plotted on the vertical axis. When the gray node sends a packet, it is forwarded to the left entity along the steepest gradient.

Algorithm 1 Steepest gradient routing

$x =$ source node
$\mathcal{S} = \{$all neighbors of $x\}$

while (there is a $y_i \in \mathcal{S}$ with $\varphi(y_i) > \varphi(x)$) **do**
$x = y_k | \varphi(y_k) = \max\{\varphi(y_i)\}, \forall y_i \in S \cap y_i \neq y_k$ {//break ties randomly}
$\mathcal{S} = \{$all neighbors of $x\}$
end while

The detailed routing algorithm is described in Algorithm 1. The steepest gradient at a node is determined by evaluating the potential values of the node's neighbors. That is, the link from a node to the neighbor with the highest potential value points to the steepest ascent. Therefore, when a node x receives a packet, it compares the potential value of all its neighbors and forwards the packet to the neighbor with the highest potential value. If the same maximum value exists at multiple neighbors, one is chosen randomly. This procedure is repeated until the packet reaches a node at a field maximum (i.e., the potential value of all neighbors is smaller than the own one). Ideally, this node is an entity and the packet has reached its destination. Otherwise, the packet is stuck in a local maximum (a maximum in the field where there is no entity). We will discuss later in this chapter how local maxima can form and how to handle them. For now let's just assume that local maxima are rare, and that we know how to route around those.

3.1.4 Forwarding with Link Failures

In Algorithm 1, we assumed that all the links are constantly

Algorithm 2 Forwarding algorithm at nodes

x = node with data packet
$\mathcal{S} = \{$neighbors of x with $\varphi > \varphi(x)\}$
bool transmitted = false

while (transmitted == false $\cap$ $S \neq \emptyset$) **do**
  $y = y_k | \varphi(y_k) = \max\{\varphi(y_i)\}, \forall y_i \in S \cap y_i \neq y_k$ {*//break ties randomly*}
  if (sendTo(y) ==successful) **then**
    transmitted = true
  else
    // link is broken
    $S = S - y$
  end if
end while

available. However, this might not be the case in wireless ad hoc networks, when nodes are moving or links could be temporarily unavailable due to for example fading or interference. Indeed, field-based routing can cope very easily with broken links as packet forwarding is based only on local decisions. Field-based routing manages to deliver packets successfully as long as one positive gradient exists between the source and destination.

In Algorithm 2, we show how nodes manage to route packets around broken links. Nodes keep track of all the neighbors with a larger potential value than they have. When they forward a packet, they first try to forward it to the neighbor with the largest potential value (along the steepest gradient). If this fails[2], they assume that the link is broken and try to forward the packet to another neighbor, namely the one with the next highest potential

[2]This failure could for example be detected with a notification from the MAC layer when the sender did not receive any acknowledgement back after a certain amount of retransmissions.

value. This procedure is repeated until a link works or until there are no longer neighbors with a larger potential value. In the latter case, the packet can no longer be delivered with this technique, and we drop the packet.

3.2 Example Scenarios

The potential field distribution and thus the routing is influenced by three factors: the capacity Q of the entities, the arrangement of the entities, and the exponent value k. To show the impact of these three factors, we discuss in the following simple example scenarios.

3.2.1 Potential Function Resulting from a Single Entity

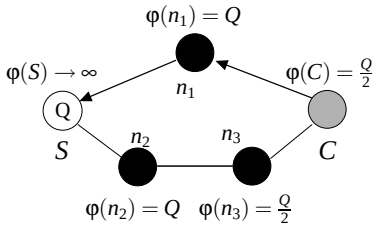


Figure 3.3: *A potential function resulting from a single entity S with capacity Q. The path traversed by a packet from C is marked by the arrows.*

We first look at the trivial case of a potential field from a single entity. Consider for example the network scenario as illustrated in Figure 3.3. A capacity Q is assigned to entity S. This capacity creates a potential field according to Equation (3.2). The potential values at all nodes are plotted for $k = 1$: The potential

value of a node is then Q over the distance to S. Note that the potential value of S is $\varphi(S) \to \infty$ since $d_S(S) = 0$.

Now assume that node C wants to send a packet to S. C has two neighbors n_1 and n_3 with potential values $\varphi(n_1) = Q$ and $\varphi(n_3) = \frac{Q}{2}$ respectively. Therefore, the packet is sent to n_1. Node n_1 is a direct neighbor of S and forwards itself the packet to S, since S has the largest potential value among all neighbors of n_1.

Note that the packet travels along the shortest path. The alternative path to S has one more hop. Generalizing this observation, we claim the following:

Theorem 1 *In a potential field with a single entity, a packet from any node in the network is routed along the shortest path to the entity for any* $Q > 0$ *and* $k > 0$.

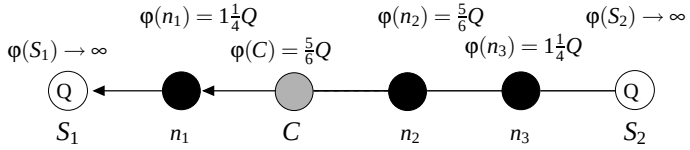
Proof: We prove this by showing that the potential value of the first hop on the shortest path is larger than any other neighboring node, and thus on the steepest gradient. Let n be the source node that sends a packet to entity node j with capacity Q. Then, assume that node x is the next hop on the shortest path between n to j. Consider an arbitrary neighbor y of node n such that $x \neq y$. Since x is on the shortest path, we claim that

$$d_j(n) = 1 + d_j(x) < 1 + d_j(y)$$

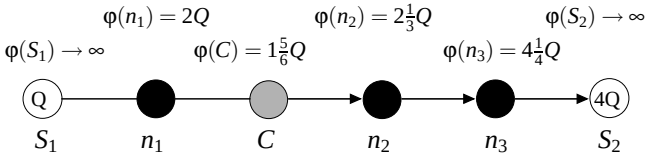
This implies that

$$d_j(x) < d_j(y)$$

Since j is the only entity, the potential value at node x is $\varphi(x) = \frac{Q}{d_j^k(x)}$ and $\varphi(y) = \frac{Q}{d_j^k(y)}$ at node y. If $d_j(x)$ is smaller than $d_j(y)$, then $\varphi(x)$ must larger than $\varphi(y)$ for any $Q > 0, k > 0$. Therefore, a packet from n or any node in the network is always forwarded along the shortest path. ■



(a) Identical capacity values.



(b) Different capacity values.

Figure 3.4: *Potential field distribution with two entities.*

3.2.2 The Effect of the Capacity Q

In a potential field with multiple entities, data packets are delivered to any entity. In general, there are many reasons to prefer a close entity over a more distant one. For example, localized communication generally reduces end-to-end delays or the probability of route failure due to mobility. However, in the presence of more distant entities with higher capacities (e.g., a more performant service or a service which is currently less utilized), it is often reasonable to access more distant entities. We next show how field-based routing can be used for this.

Figure 3.4(a) shows an example network with two entities having the same capacity Q. A sender C is two hops away from entity S_1 and three hops away from entity S_2. The potential values at the nodes are calculated according Equation (3.3) with $k = 1$ and specified in the figure. As we see, the steepest gradient at node C is directed towards the closer entity S_1. In Figure 3.4(b), the topology remains the same except that this time, the capacity of S_2 is increased to $4Q$. This difference causes the steepest

gradient to be now directed towards the farther entity S_2.

As we see, by increasing its capacity, an entity can influence the potential field in its favor; the higher the capacity an entity uses, the higher the surrounding potential values are, and the more the number of packets that are attracted to this entity.

3.2.3 The Effect of Entity Concentration (Density)

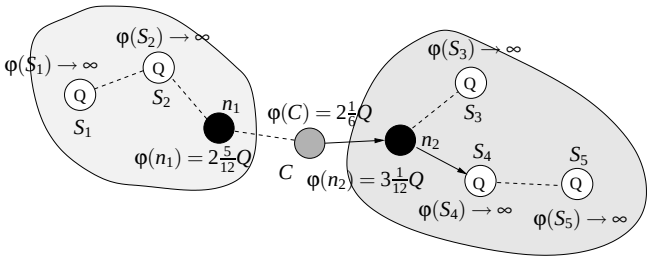


Figure 3.5: *A potential function resulting from five entities ($S_1 - S_5$) having the same capacity Q. The steepest gradient at node C is directed towards the largest entity concentration.*

Another property of field-based routing is its ability to route packets towards regions in the network which are highly populated by entities. This is what we refer to as density-based routing as we will see in the next chapter. Density-based routing is particularly useful in dynamic networks such as for example mobile ad hoc networks. There, the probability of successful packet delivery is higher when a packet is routed in the direction of many entities. This is mainly because we can re-route a packet to an alternative entity if the targeted one, or links to it, become unavailable.

This property is best illustrated in Figure 3.5 for values of $k = 1$. Five entities ($S_1 - S_5$) having the same capacity Q are located around a sender C. Entities S_2, S_3, and S_4 are all three at a distance of two hops from C. A shortest path routing algorithm

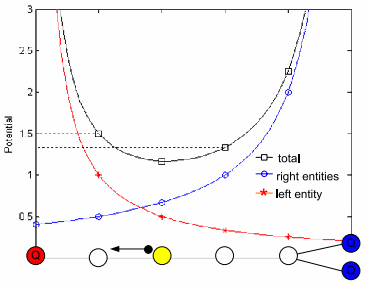
would thus treat them as equally optimal. However, with field-based routing the steepest gradient is directed towards the right portion of the network with the highest entity concentration. This is due to the summation of the capacities and the consequentially higher potential in that direction.

This example illustrates the benefit of using field-based routing in dynamic environments. Assume that C sends a packet towards S_2. If S_2 moves away or just disappears before the field values can be updated by the protocol, the packet will be dropped at the relaying node. Now assume that C sends its query towards S_3 which in turn disappears. The intermediate node between C and S_3 is able this time to react locally and forward the packet to an alternative entity (in this case S_4), and successfully deliver the packet.

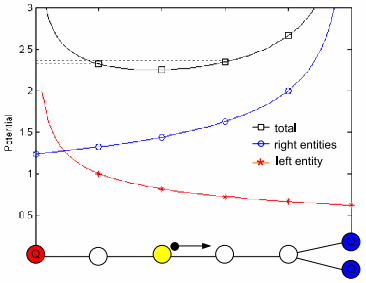
3.2.4 The Effect of k

The exponent k in Equation (3.3) also determines how packets are routed in the network. Large values of k result in steep potential fields whereas small values of k result in flatter distributions.

This effect is best illustrated in Figure 3.6. Node C is two hops away from the left entity and three hops away from the two entities on the right. The potential field for this topology is plotted for $k = 1$ and $k = 0.3$ in (a) and (b) respectively. For $k = 1$, the steepest gradient at node C is towards the left. For $k = 0.3$, the steepest gradient is in the opposite direction towards the right. From this example, we see how k affects the way proximity and density is weighted in the routing decisions. This effect is studied in more detail in Chapter 4.



(a) k=1



(b) k=0.3

Figure 3.6: *The potential field distribution for different values of k.*

3.3 Analysis

In the following, we analyze field-based routing analytically and with simulations. We analyze loop-freeness and convergence for the static case (no mobility, no link failures). Route convergence in the dynamic case is considered in the next two chapters when field-based routing is applied to anycast routing and service discovery in wireless ad hoc networks.

3.3.1 Loop-freeness

An important characteristic of a routing algorithm is to provide loop-free routes. We provide the following proof sketch that field-based routing is loop-free.

Theorem 2 *Field-based routing is loop-free for any* $Q > 0, k > 0$.

Sketch of Proof: Gradient-based routing along the steepest ascent requires that the potential value at every hop on a path is strictly larger than at the previous hop. In a loop, a packet would have to traverse one node more than once which is not possible per definition. ■

3.3.2 Convergence

Since packets are routed along the steepest ascent, gradient-based routing only manages to deliver data packets to an entity if there are no local maxima (a local maximum is a node which has a higher potential value than all its neighbors but is not an entity) in the potential field. While this property is given in physics (gravitation fields or electric fields never have local maxima), in our model local maxima may occur. The reason is that potential fields in physics are continuous whereas in our model they are only defined at the network nodes.

To better understand the effect of local maxima in networks, we first give a typical scenario where a local maximum occurs. Then, we derive a lower bound for k for which we prove that local maxima never occur. Finally, we analyze the frequency of occurrence of local maxima in random network topologies for the whole range of k and show that the probability of occurrence of local maxima is low.

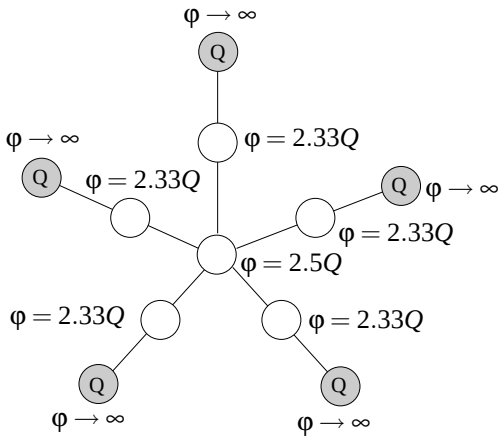


Figure 3.7: *Local maximum.*

Example for Local Maximum

Local maxima typically occur in specific star topologies such as for example in Figure 3.7. Five entities are situated at the endpoints of a star. The potential value for $k = 1$ at the central node is $\varphi = 2.5Q$ which is larger than all its direct neighbors ($\varphi = 2.33Q$). Note that the occurrence of a local maximum at the central node depends on the value of k. In this example, for $k > 1.4946$, the local maximum disappears.

Lower Bound for k

We have seen that the occurrence of local maxima depends on the value of k. Next, we derive a lower bound for $k > k_b$ such that we can exclude the occurrence of local maxima.

Lemma 3 *Consider a connected network with diameter*[3] *D and*

[3]Note that the network diameter D in this thesis is defined as the longest shortest path between any pair of nodes in the network.

with an entity group of size N. The potential function shows no other maxima than at nodes which are group members, if k is any constant satisfying:

$$k > \frac{\log N}{\log \frac{D}{D-1}} \tag{3.4}$$

Proof: Consider any node x which is not member of the group. To guarantee that the potential value at x is not a maximum, there must at least be one neighbor y with a greater potential value:

$$\varphi(x) < \varphi(y) \tag{3.5}$$

We denote $\mathcal{N}$ as the set of group members and assume that the distances from x to all group members $(d_i, \forall i \in \mathcal{N})$ are known. Then we can calculate $\varphi(x)$ and $\varphi(y)$:

$$\varphi(x) = \sum_{i \in \mathcal{N}} \frac{1}{d_i^k} < \sum_{i \in \mathcal{N}_1} \frac{1}{(d_i - 1)^k} + \sum_{i \in \mathcal{N}_2} \frac{1}{d_i^k} + \sum_{i \in \mathcal{N}_3} \frac{1}{(d_i + 1)^k} = \varphi(y) \tag{3.6}$$

where $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ are disjoint subsets of $\mathcal{N}$ ($\mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3 = \mathcal{N}$ and $\mathcal{N}_u \cap \mathcal{N}_v = \emptyset, \forall u, v = 1, 2, 3; u \neq v$). The potential value of y must be of this form since x and y are direct neighbors, and the distance of y to any group member can only be one hop smaller, equal, or one hop larger than the distance of x to the corresponding member. Since all terms in Equation (3.6) are positive by definition, if the potential value at y is still larger than the potential value at x when only considering the contribution from the group members $\mathcal{N}_1$, the potential at node x cannot be a local maximum. Thus, we can simplify the previous condition to

$$\varphi(x) = \sum_{i \in \mathcal{N}} \frac{1}{d_i^k} < \sum_{i \in \mathcal{N}_1} \frac{1}{(d_i - 1)^k} \leq \varphi(y) \tag{3.7}$$

We can even further simplify by only considering at y the potential value from the closest group member s from x:

$$\varphi(x) = \sum_{i \in \mathcal{N}} \frac{1}{d_i^k} < \frac{1}{(d_s - 1)^k} \tag{3.8}$$

We now consider the worst case potential value of x. The worst case value is when the potential is maximal. The maximal potential value for x is obtained when the distances to the group members are minimal. Since we said that s is the closest member, d_s is the smallest distance that any group member can have. Therefore, the worst case is when all group members are at distance d_s:

$$\varphi(x) \leq \frac{N}{d_s^k} < \frac{1}{(d_s - 1)^k} \tag{3.9}$$

We now solve for k and get

$$k > \frac{\log N}{\log \frac{d_s}{d_s - 1}} \tag{3.10}$$

The distance d_s is strictly smaller than the network diameter D and since $\log \frac{d_s}{d_s - 1} > \log \frac{D}{D-1}$, the condition that the potential value of x is smaller than any neighbor node y is

$$k > \frac{\log N}{\log \frac{D}{D-1}} \tag{3.11}$$

■

We identified the range of k where we can guarantee that there are no local maxima and thus guarantee that routing converges. However, as we will see later, density-based routing strategies require a smaller value of k. Consequently, we also use our model for smaller values of k than the derived bound. When k is smaller than the derived bound, there is no strict guarantee that a packet will reach its destination since local maxima may occur in the potential field. However, we show next that local maxima occur

extremely infrequently for such values in random networks. Note also that additional protocol mechanisms may be used in such cases as local maxima can easily be detected locally.

Simulations with Random Networks

We now show with simulations of static random networks that the occurrence of local maxima is rare for any $k < \frac{\log N}{\log \frac{D}{D-1}}$. The random networks we use to calculate the potential fields are generated in the following way. Nodes are placed at random positions on a square. There exists a link between two nodes if their geometric distance is smaller than a threshold value (corresponding to the wireless range of the radio devices).

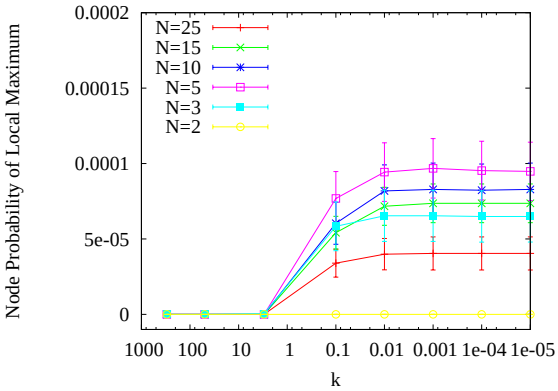


Figure 3.8: *Local maximum probability versus k.*

Figure 3.8 shows the probability that a non-member node is a local maximum for values of k between 1000 and 10^{-5} with a 90 percent confidence interval. The resulting curves are obtained by averaging at least two thousand random networks with 500 nodes. The average node degree in those networks is approximately 12, and the average network diameter around $D = 20$. The plot shows

that for large values of k ($k > 3$), there are no local maxima in the potential field independent of the group size. For $k < 3$, few local maxima form. However, the probability remains quite low. In this specific case, the probability is always below 0.01% for the used group sizes.

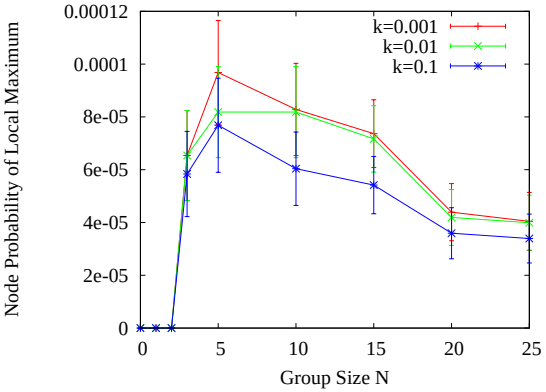


Figure 3.9: *Local maximum probability versus group size N.*

We also plot the probability of local maxima versus the group size on the horizontal axis in Figure 3.9. For group sizes of 2 and below, there are no local maxima. Local maxima start to happen at a group size of 3. Note, that the highest probability for a local maximum is reached with a group size of 5. For larger groups, the probability decreases significantly. This is mainly because local maxima form only in specific star topologies, when the group members are in different directions but about the same distance from a node. By increasing the group size, the probability that such a topology occurs, decreases.

In Figure 3.10, the occurrence of local maxima is plotted for different network densities. We express the network density with the average node degree of the nodes on the horizontal axis. We obtained networks of different average node degrees (densities) by

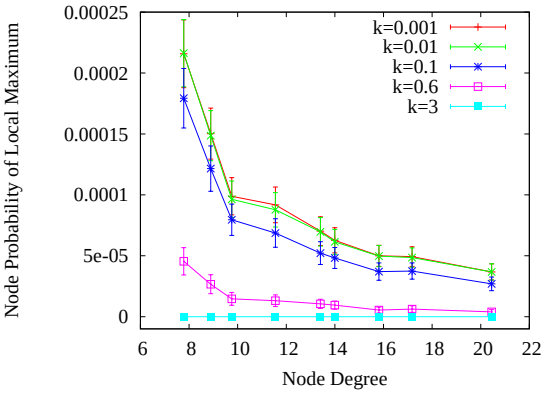


Figure 3.10: *Local maximum probability versus network density.*

varying the number of nodes while keeping the simulation area size constant. For the smallest average node degree of 7.8, the number of nodes was 250 and for the largest average node degree of 20.5, the number of nodes was 700. We observe a higher probability for local maxima in low density networks. For example, with $k = 0.1$, the probability of a local maximum at a node is 0.018% for a node degree of 7.8. However, by increasing the node degree to 20.5, the probability drops to 0.0027%. This result confirms our expectation. By increasing the node density in our model, the discrete field approaches a continuous field without any constraints in the propagation space (which does not show local maxima). As previously mentioned, for $k \geq 3$, there are no local maxima in the field.

We conclude that the probability of having local maxima in random networks is low.

3.4 Conclusions

In this chapter, we presented field-based routing, a routing model inspired from fields in physics. Field-based routing is robust: it can handle broken links locally by rerouting packets over any available links as long as the gradient over those links is ascending.

When there is only one entity in a group, field-based routing manages to deliver packets over the shortest route to that entity. When multiple entities contribute to a potential field, it performs anycast which is affected by the capacities at the entities, the density of the entities, and the value of k.

With a detailed analysis, we show that field-based routing is loop-free and it converges (there are no local maxima) for $k > \frac{\log N}{\log \frac{D}{D-1}}$. For smaller values of k, we show through simulations that the probability for a local maximum is very low. Therefore, we regard those cases as exceptions, and consider using field-based routing for smaller values of k.

Chapter 4

Density-based Anycasting

In this chapter, we introduce a novel anycast routing strategy: density-based anycasting. Existing anycast routing protocols (which we will call "proximity-based routing protocols") solely route packets to the closest group member. Density-based routing further considers the number of available group members in the routing direction. Routing based on density is particularly useful in mobile networks, where the closest group member can move away, or when the path to it can break.

We explore density-based anycast routing with the field-based routing model we presented in the previous chapter. By shaping the field using different values of k (see Equation 3.3), we are able to express pure density-based routing strategies, but also pure proximity-based strategies, and combined routing strategies. Using this unified model, we compare the performance of the different strategies in two specific scenarios: a mesh network and a sensor network. Our main findings are that density-based anycasting increases the probability of successful packet delivery when the network topology can change frequently, and it even

finds shorter routes compared to proximity-based routing when the network topology significantly changes during the transmission of data packets.

4.1 The Need for Density-based Anycasting

Anycast is a delivery mode in which a sender sends a packet that is to be delivered to any member from a group. The specific member which receives the packet is not predetermined at the sender. Rather, the sender specifies the group of potential receivers and the network routes packets to any member that is appropriate according to the routing metric. Traditional protocols [?, ?, ?, ?, ?] are all based on unicast routing techniques such as e.g., distance vector or link state routing. As a result, data packets are solely based on proximity information.

We argue that there is a need for routing based on density information in dynamic networks such as mobile or ad hoc networks. The reason is that it is easier to deliver packets successfully to a member in regions of high member densities than in regions with low member densities.

4.2 Anycast Routing Model

We use the field-based routing model to explore different strategies for anycast. For simplicity, we assume that all members of a group are equivalent and set the capacity for all members of the group to the same value $Q_j = 1, \forall j \in \mathcal{N}$ (Note that we explore the case of diverse capacities in the next chapter when applying field-based routing to service discovery).

The particular potential field and thus also the resulting routing strategy is selected in our model via the exponent k. We start

by deriving bounds for k where we can guarantee that the resulting routing strategy is pure density-based or pure proximity-based.

4.3 Analytical Evaluation

4.3.1 Bound for Pure Proximity-based Routing

Lemma 4 *Consider a network with diameter D and an anycast group of size N. A packet from any node in the network is always routed to the closest group member of the anycast group along the shortest path, if k is a constant satisfying*

$$k > \mu(N, D) \tag{4.1}$$

(see Table A.1 in the appendix for typical values of μ as it cannot be expressed analytically)

Proof: We prove this by determining the steepest gradient in a worst case scenario and show that it points to the shortest path. The worst case scenario is obtained when all members of a group are located in the opposite direction of the closest group member. This scenario is depicted in Figure 4.1. Note that only the nodes which are relevant to the proof are shown. The network topology might as well include any number of additional nodes.

The node m_s is the closest group member to the sender, however, the remaining $N-1$ group members $(m_{l1}, ..., m_{ln})$ are all located at the opposite end. This arrangement is the worst case because $m_{l1}, ..., m_{ln}$ create the largest possible potential at node n_2 which is in the opposite direction of the closest member, and the smallest possible potential at n_1 which is on the shortest path to m_s. If we can guarantee that $\varphi(n_1)$, which is the potential of the node on the shortest path to m_s, is larger than $\varphi(n_2)$, then a packet following the steepest ascent will be forwarded to n_1 towards m_s on the shortest path. Therefore, the following condition

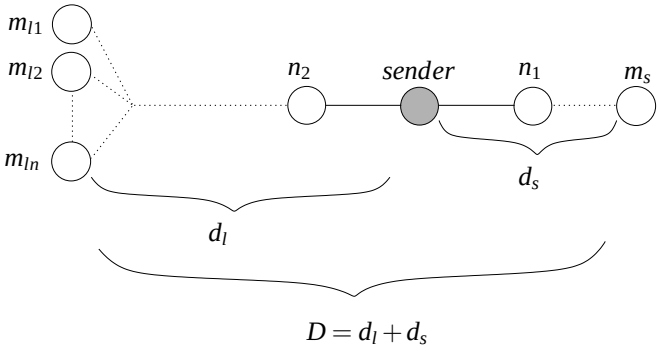


Figure 4.1: *Network topology with one member at distance d_s and $N-1$ members at distance $d_l > d_s$.*

must be met

$$\varphi(n_1) > \varphi(n_2)$$

We assume that $m_{l1}, ..., m_{ln}$ are all equidistant with distance d_l from the sender, and we calculate the potential values using Equation (3.3)

$$\varphi(n_1) = \frac{1}{(d_s-1)^k} + \sum_{N-1} \frac{1}{(d_l+1)^k} > \frac{1}{(d_s+1)^k} + \sum_{N-1} \frac{1}{(d_l-1)^k} = \varphi(n_2) \tag{4.2}$$

This inequality holds in all cases, if it also holds for the smallest value of d_l, and the largest value of d_s. Therefore, we set the distance d_l to the smallest possible distance $d_l = d_s + 1$ (which is one hop more than the distance to the closest member m_s) and d_s to the largest possible value $d_s = \frac{D-1}{2}$ (since $D = d_s + d_l$) and get

$$\frac{1}{(\frac{D-1}{2}-1)^k} + \frac{N-1}{(\frac{D-1}{2}+2)^k} > \frac{1}{(\frac{D-1}{2}+1)^k} + \frac{N-1}{(\frac{D-1}{2})^k}. \tag{4.3}$$

This equation cannot be solved analytically for k. We therefore bring it into a cancellation-free form to be able to solve it numerically (see Appendix A.1 for further details how we obtained this transformation):

$$f(k) > N - 1 \tag{4.4}$$

with

$$f(k) = \left(\frac{\frac{D-1}{2}(\frac{D-1}{2}+2)}{(\frac{D-1}{2}+1)(\frac{D-1}{2}-1)} \right)^{\frac{k}{2}} \cdot \frac{\sinh\left(\frac{k}{2}\log(\frac{\frac{D-1}{2}-1}{\frac{D-1}{2}+1})\right)}{\sinh\left(\frac{k}{2}\log(\frac{\frac{D-1}{2}}{\frac{D-1}{2}+2})\right)}. \tag{4.5}$$

The solution for sample values of D and N is given in Table A.1. Since for $k > \mu$, the steepest gradient is always pointing to the shortest path, a packet from any node in the network is always routed along the shortest path. ■

4.3.2 Bound for Pure Density-based Routing

We now derive the other limit in which a routing strategy always forwards packets towards the highest member density and the distance of a node to group members is irrelevant.

Lemma 5 *For a network with diameter D and an anycast group of size N, a packet from any node in the network is routed independently of the distances to the members if k is any constant satisfying*

$$0 < k \leq \varepsilon(D,N) \tag{4.6}$$

and

$$N > \frac{\log 3}{\log \frac{D-1}{D-3}} + 1 \tag{4.7}$$

(see Table A.2 in the appendix for typical values of ε as it cannot be expressed analytically)

Proof: We prove this by calculating the steepest gradient in a best case scenario and show that it does not point towards the next hop on the shortest path to the closest member. We consider again the notation from Figure 4.1 where the closest group member m_s is at distance d_s from a sender. We determine the smallest factor k when a packet is not routed towards the closest group member independent of how close it is to this member. If a packet is not routed along the shortest path, then the potential value of the next hop towards this member must be smaller or equal to at least one other neighbor

$$\varphi(n_1) \leq \varphi(n_2) \tag{4.8}$$

or

$$\frac{1}{(d_s-1)^k} + \sum_{N-1} \frac{1}{(d_l+1)^k} \leq \frac{1}{(d_s+1)^k} + \sum_{N-1} \frac{1}{(d_l-1)^k}. \tag{4.9}$$

In the best case, m_s is very close to the sender and all other group members are far away. Therefore, we set d_s to the smallest possible value $d_s = 2$ and d_l to the largest possible value $d_l = D - d_s = D - 2$ (Note that we do not consider the trivial case $d_s = 1$ since then a packet will always be routed to m_s because it is now a direct neighbor of m_s with a potential of $\varphi \to \infty$). Now we get

$$1 + \frac{N-1}{(D-1)^k} \leq \frac{1}{3^k} + \frac{N-1}{(D-3)^k}. \tag{4.10}$$

This inequality cannot be solved analytically for k and we therefore bring it into a cancellation-free form to solve it numerically (see Appendix A for further details)

$$f(k) \leq N-1 \tag{4.11}$$

with

$$\left(\frac{(D-1)(D-3)}{3}\right)^{k/2} \cdot \frac{\sinh\left(\frac{k}{2}\log 3\right)}{\sinh\left(\frac{k}{2}\log(\frac{D-1}{D-3})\right)}. \tag{4.12}$$

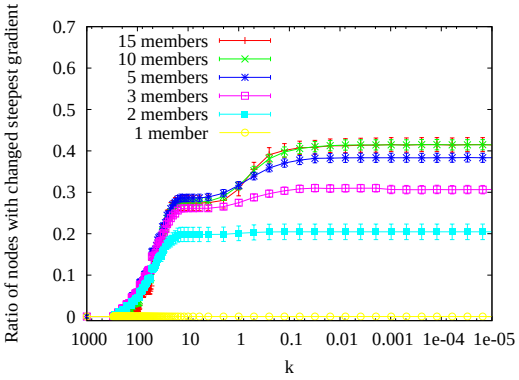
Note that this inequality has solutions only for $N-1 > \frac{\log 3}{\log \frac{D-1}{D-3}}$. The solution of this inequality is given for sample values of D and N in Table A.2. Since for $k \leq \varepsilon$, the steepest gradient does not correlate with the shortest path, packets are routed independently of the distance to the members, namely based on density. ■

4.4 Simulation Study

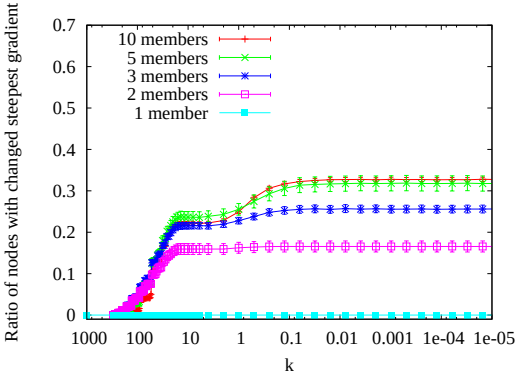
In the previous section, we identified upper bounds for k when the anycast routing strategy always finds the shortest path and strategies where the member distance has no effect on packet delivery. In the following, we study the routing behavior of routing strategies with values of k between these two extreme cases.

4.4.1 Effects of k on Potential Field

To determine the effect of different values of k on the steepest gradient of the potential field, we do the following. We calculate the potential field for a network with a very high value of k (in this case $k = 1000$) as a reference field. This high value virtually guarantees that the resulting steepest gradient is pointing to the next hop along the shortest path to the closest member at every node (see Section 4.3.1 for the proof). Then, we calculate the potential field for different values of k with the same network topology, and compare these with the reference field. We define the metric for the difference of two fields as the number of nodes at which the direction of the steepest gradient has changed. For example, a difference of 0.5 means that at 50% of the nodes, the neighbor with the highest potential value is not identical compared to the node of the reference field. Note that since the distance metric indicates how often the steepest gradient changes in a field, it also captures the effect of changes in the routing strategy because packets are routed along the steepest gradient.



(a) diameter D=18.



(b) diameter D=13.

Figure 4.2: *Effect of k on the steepest gradient.*

In the following, we plot the difference of the reference field to fields obtained with various values of k. The plots are obtained by averaging the result of random wireless networks, which were generated as follows. A fixed number of nodes are randomly placed on a quadratic plane (2000m by 2000m). There exists a bidirectional link between two nodes if their geometric distance is less than, or equal to, the wireless range (a fixed value identical for

all nodes). This model corresponds to a flat environment with devices equipped with wireless radio, all having equal communication range. The anycast group members are assigned randomly to the existing nodes.

In Figure 4.2(a), we used a network size of 400 nodes and set the wireless radio range to $200m$ in order to obtain a connected network (there exists a path between any two nodes). All networks had a network diameter size of $D = 18$ (which is the most likely network diameter for the given parameters) and the average node degree was approximately 12. Group sizes between 1 and 15 members were used. For the different group sizes, we plot the ratio of nodes affected by changes in the field on the vertical axis for values of k between 1000 and 10^{-5} with a 90 percent confidence interval. For all group sizes, the potential field differs from the reference field for values of k below approximately 200. When k becomes smaller than approximately 0.01, reducing k further does not produce any change in the field. We also observe that the difference between the reference field is larger for larger groups. For a group size of 15 members, the steepest gradient changes at more than 40% of the nodes. Note that the steepest gradient is not affected by k for a group size of one member. In that case, the potential field always results in a shortest path routing strategy for all values of k as proved in Section 4.3.1.

In Figure 4.2(b), we reduced the number of nodes to 200 and increased the wireless range to 280m to still obtain connected networks (if a generated topology was not fully connected, we discarded it). All networks were of diameter $D = 13$ and the average node degree was approximately 12 as in the previous plot. The resulting plot is similar to Figure 4.2(a). The steepest gradient changes occur at similar values of k for the respective group sizes. For small values of k however, the total field difference is smaller.

We also investigated the effect of k on the field for different network densities. In Figure 4.3, the difference using a 90-percent

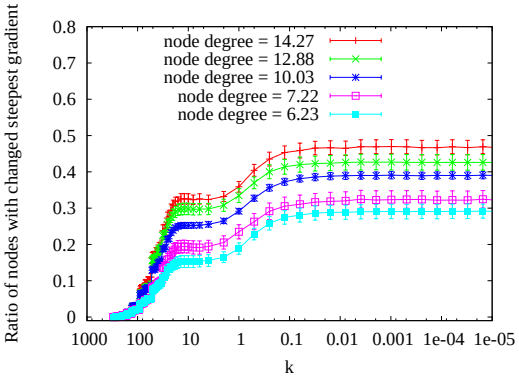


Figure 4.3: *Effect of k on the steepest gradient.*

confidence interval is plotted for different network densities and a fixed group size of $N = 10$. Different network densities were obtained by varying the number of nodes from 220 to 500 while keeping the wireless range of each device to $200m$. We measure the network density as the node degree averaged over all nodes. As the plot shows, the potential field is subject to more changes when the network density is high. This is due to the fact that the denser the network, the large the number of available paths.

4.4.2 Four Routing Categories

Category	Proximity-based routing	Density-based routing	Delivery to closest member over shortest path
I	yes	no	yes
II	yes	yes	yes
III	yes	yes	no
IV	no	yes	no

Table 4.1: *Routing categories and their properties.*

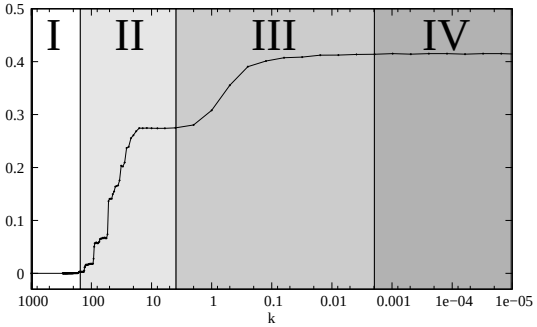


Figure 4.4: *Four categories of strategies (curve for D=18, N=15).*

The shape of the curves in Figure 4.2 and 4.3 show clear patterns and we use them to classify the routing strategies into different categories. In the following, we define four main categories (I - IV) of routing strategies and explain their properties.

We know from Section 4.3.1 that for very large values of k, packets are routed to the closest member without considering the member density in any specific direction. Therefore, if we look at the field difference curve from Figure 4.4 (this example curve was obtained with D=18 and N=15), we can classify all routing strategies with no difference compared to the reference field, as strategies that only consider the closest group member for routing decisions. We marked these strategies with category I in the figure. Note that existing anycast routing protocols ($[?, ?, \mathbf{?}, ?, ?]$) all employ such a proximity-based routing strategy.

For k smaller than approximately 200 in the figure, the influence of density starts to act since the difference to the reference field is greater than zero. However, this shift in the steepest gradient does not impact the delivery path length since we know from the previous section that even in the worst case, in a network with diameter D=18 and group size N=15, as used in the

example, any strategy with a value of $k > 20$ still forwards over a shortest route. What happens in this case is that when different members are equidistant and closest to a node, the member which is the direction of the highest member density is preferred. We define this range as a separate category (category II) of routing strategies, where packets are still routed to one of the closest member over the shortest path but where density is considered at the same time. We will see later in this section the benefits of category II versus category I in terms of robustness.

By further decreasing k, the effect of density-based routing outweighs more and more the effect of proximity-based routing in the routing strategy. At a certain point, packets start to travel over paths which are no longer the shortest ones. As soon as one delivery path in the network is no longer the shortest, we declare the routing strategy to belong to the category III. Thus, in this category, routing decisions are based on a tradeoff between member distance and member concentration, but in contrast to category II, packets are no longer delivered over the shortest paths to the closest member.

For very small values of k (smaller than 0.001 in the example), the plot shows that the potential field no longer changes when decreasing k. Therefore, we define category IV at this point. With category IV, routing decisions are only based on the member concentration and the member distance is no longer relevant. Note that the limit between phase III and phase IV is not as sharp as indicated in the figure. The limit could be as well more to the left or to the right side. However, we did not find an easy way to determine the transition range. To summarize, the four categories and their properties are listed in Table 4.1.

To highlight the transition line between category II and III, we further plot the transition value of k between these two categories versus the group size N in a separate plot in Figure 4.5. Recall that the difference between category II and III is that packets in category III are no longer routed over the shortest path,

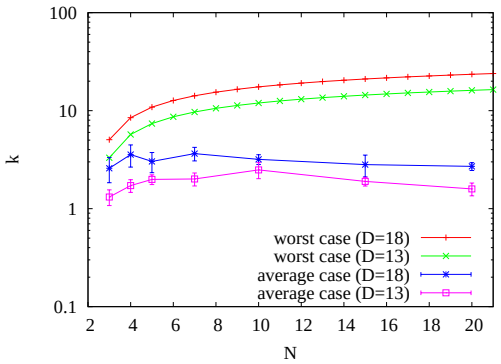


Figure 4.5: *Transition between category II and III. The value of k indicates the smallest value that still results in shortest path forwarding. The worst case is the derived upper bound, and the average case is an empirical value obtained with simulations.*

which is still the case in category II. Thus, this value is the smallest possible value of k until routing is still following the shortest path. For completeness, we also plot the worst case bound μ obtained in the previous section. We find that the values obtained with the simulations (denoted with average case values) on random networks with $D = 13$ and $D = 18$ are between $4 > k > 1$. The worst case values are quite close to the average values for small group sizes, and tend to be significantly larger for group sizes around 20 members. Note that in contrast to the worst case values, the transition limit between the two categories does not increase for larger group sizes.

4.4.3 Evaluation with Mesh Network Scenario

In this subsection, we evaluate the different anycast routing strategies in a mesh network scenario. A mesh network is a wireless network of fixed nodes that provides Internet connectivity to mobile

users. For economic reasons, not all the fixed nodes have an Internet connection but only a subset of them. The nodes that do not have a direct connection to the Internet thus forward their traffic over radio to any mesh node which has an Internet connection. Anycast routing is a natural fit to route packets in a mesh network since the packets can be sent to the Internet via any node.

Early experiences with mesh networks [?] showed that although the nodes are fixed, wireless links between the nodes may appear and disappear (mostly due to multipath fading effects). In the following, we analyze the performance of the different routing strategies in such a mesh network scenario.

Simulation Model

We generate random networks (in the same way as in the previous subsection) with 500 nodes from which ten are randomly selected as group members (fixed mesh nodes with Internet connection). For each network, we establish potential fields using different values of k. We model link instabilities by declaring links unavailable (no packets can be sent at all) after the potential field is setup. These failures cause links to become unavailable that are located along the steepest gradient. When an intermediate node wants to forward a packet via a link that is unavailable, it forwards packets to the neighbor with the next highest potential value. If it has no direct neighbor with a larger potential value than its own potential values, it drops the packets.

Simulation Results

The metric we use to compare the different routing strategies using the different potential fields is the packet delivery ratio. We count the number of data packets that were received at group members versus the total number of sent data packets. Recall

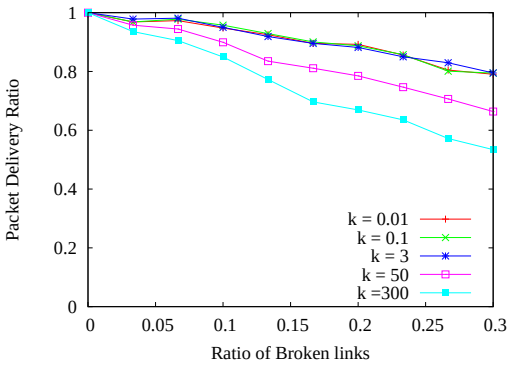


Figure 4.6: *Packet delivery ratio in mesh networks for different routing strategies.*

that packets are dropped when they reach a node which lost all its links with an increasing gradient.

The packet delivery ratio for this experiment is plotted in Figure 4.6. On the horizontal axis, we plot the ratio of broken links which indicates the degree of network stability/instability. A value of 0.1 means that 10% of the existing links in the original network topology are unavailable.

We observe that when the ratio of broken links is zero, the packet delivery ratio is 100% for all anycast routing strategies. When the ratio of broken links increases, the packet delivery ratio is better for small values of k (i.e., the case that packets are routed based on member density). For example, when 30% of all links are broken, still 80% of the packets are delivered for $k < 3$ whereas only 66% of the packets are delivered for $k = 50$, and 53% of the packets for $k = 300$. Notice that the packet delivery ratio in this scenario is roughly the same for $k = 3$, $k = 0.1$, and $k = 0.01$.

This is an interesting result since for $k = 3$, the routing strategy is in phase II, which means that packets are still forwarded over the shortest path. That is, with $k = 3$ we profit from the ro-

bustness of density-based routing without the penalty of longer paths.

we will always forward packets over the shortest path. In dynamic networks, the density-based routing however, increases the packet delivery rate by 23% compared to pure ($k = 300$) proximity-based routing strategies.

4.4.4 Evaluation with Sensor Network Scenario

In the following, we evaluate the performance of different anycast routing strategies in a sensor network scenario. Sensor networks are used to collect environmental data from sensors. A typical sensor network consists of small, autonomous, battery-powered sensor nodes connected via radio links. The data collected by these sensors is then transmitted over multiple hops to the closest data sink for further processing or analysis. Since sensors can send their data to any of the data sink nodes, anycast is a convenient delivery mode for this type of scenario. Sensor networks often have hard energy and resource constraints due to the limitation of the sensor nodes but, on the other hand, they are more delay tolerant than traditional data networks. As a consequence, sensor networks employ extensive usage of sleep operations and caching. Therefore, packets from sensors are often delayed in the network until they arrive at the destination and the end-to-end delay can thus be in the order of seconds or even minutes compared to milliseconds if it would have been sent directly. Due to the increased end-to-end delays, it is likely that in mobile scenarios, the network topology changes during the data delivery.

In this subsection, we analyze the efficiency of different anycast delivery strategies in scenarios where topology changes occur during ongoing transmissions. We do so by analyzing the delivery path length of different anycast routing strategies which we instantiate with our model by selecting specific values for k. We consider sensor networks with three different mobility patterns.

In the first scenario, the sensors as well as the data sinks are mobile. In the second scenario, the sensors are mobile but, the data sinks are at fixed locations. In the third scenario, the sensors are fixed and the data is collected by mobile data sinks.

Simulation Model

We model communication and node mobility in the following way. At time t_0, the potential field is created in the network and a random sensor node generates a data packet which is sent via anycast to the neighbor along the steepest ascent of the potential field. Intermediate nodes cache the data packet for a time period δt before forwarding the packet. After δt, the potential field is recalculated at each node according to the new network topology (if nodes moved within the delay time δt) and the packet is then forwarded again towards the steepest ascent. This procedure is repeated until the packet arrives at a data sink. Note that with this procedure, even if the nodes are mobile, the potential field is always correct when a packet is forwarded. In practice, it would require that the control messages to establish the potential fields have been sent right before the packet is sent. Although not completely realistic, this idealized scheme allows us to study the performance of the different anycast routing strategies separately without accounting for the inaccuracy of an underlying protocol used to establish the respective routing tables. Also, this idealized scheme guarantees that, unless there are local maxima in the potential field, data packets will reach a data sink. Note that for fairness in the comparison of the different strategies (some strategies have more local maxima than others as we have seen in previous evaluation), we did not account for simulations in which local maxima occurred. As discussed before, local maxima appear only very rarely (in less than 0.1% of all simulations).

In spite of its known limitations [**?**], we use the random waypoint model [?] as the mobility model. The problem with the

random waypoint model is that the node distribution changes during the simulations if the nodes are placed uniformly at the beginning. For this purpose, we place the nodes according to [**?**] to make sure that the simulations are conducted in steady state. 500 nodes move around in a quadratic area with a side length of 2000 m. With the random waypoint model, each node chooses a random destination and moves towards this destination on a straight line with a constant speed v. In this model, a node stops moving for a constant pause time p when it arrives at its destination. We set this pause time to $p = 0s$ which means that the nodes are constantly moving in the simulations. Ten of the 500 nodes are data sinks belonging to the same anycast group. The remaining nodes are all sensor nodes. The wireless range of the radio device at each node is set to 180 meters. There exists a link between two devices if their geometric distance is smaller than the wireless range. We chose the above parameters so that the network remains connected during the total simulations and we do not have to account for the case where nodes are within a cluster, unable to access any data sink. We do not model the behavior of any MAC or physical layer characteristics in the simulations as we are only interested in the traversed path length of data packets.

Results

We investigated the path length traversed by packets for the different anycast routing strategies. In Figure 4.7, the results are plotted with a 90-percent confidence interval for the first scenario where all nodes are mobile. The horizontal axis of the plot indicates the node speed multiplied with the forwarding delay δt. This factor is an indicator of how much the network topology changes between each forwarding step. The metric indicates the distance covered by a node and is measured in meters. On the vertical axis, the ranking of the different routing strategies is plot-

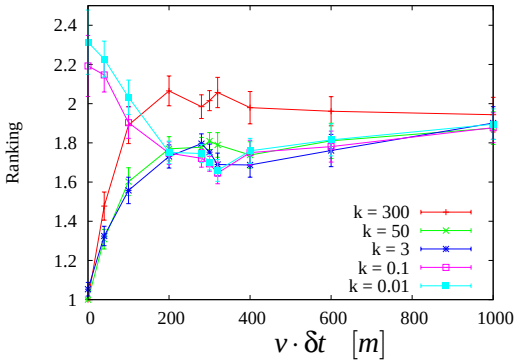


Figure 4.7: *Traversed path length with moving sensors and data sinks.*

ted. The ranking is an averaged value over all simulation runs. For example, if an anycast forwarding strategy has a ranking of 1, it means that the packets delivered with this strategy were delivered over shorter paths than the other strategies in all simulation runs. Note that we chose to plot a ranking of the strategies instead of the average path length because few paths happened to be a lot larger compared to the remaining ones which makes an average value of different runs somewhat unfair.

We find that when the node speed or the forwarding delay is zero ($0m$ on the horizontal axis), forwarding strategies with a larger value of k (strategies which favor proximity) tend to find shorter paths than strategies with small values. More precisely, strategies with $k = 300, 50$, and 3 always find shorter paths than strategies with $k = 0.1$ and 0.01. When the speed-delay product increases, we see that the performance for $k = 300$ quickly decreases compared to the strategies with $k = 50$ and $k = 3$. At a speed-delay product of $100m$ (e.g. $v = 10m/s$ and $\delta t = 10s$), the average ranking for $k = 0.1$ is even equal to the strategy with $k = 300$. At a speed-delay product of $280m$, the strategies with

$k = 0.1$ and $k = 0.01$ even find shorter paths than all other strategies which were better in the static case. When the speed-delay product becomes too large ($1000m$), the topology changes between two transmissions is so significant that the strategy has no more any effect on the performance.

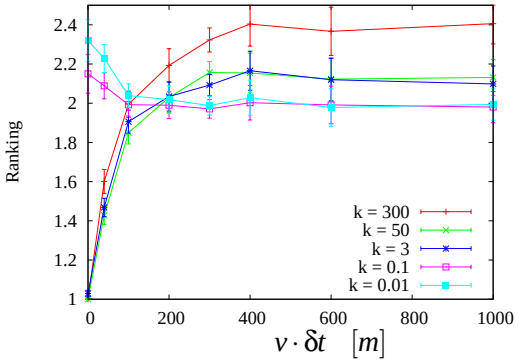


Figure 4.8: *Traversed path length with fixed data sinks and mobile sensors.*

In the next scenario (see Figure 4.8), we did the same experiment while maintaining the data sinks at a fixed (random) location. For the speed-delay product between 0 and 200, the trend is similar to the previous experiment in which all nodes were mobile. However, we see that for speed-delay products larger than 200, strategies with small k (in this case $k = 0.1$ and 0.01) tend to find shorter paths. Even when the speed-delay factor is 1000, these strategies perform better than the others.

We also analyzed the effect of k in scenarios where the sensor nodes are all at a fixed locations and the data collecting devices are moving (Figure 4.9). In this set of scenarios, the strategies with a small values of k take longer to outperform the others. However, for a speed-delay product beyond 300, the strategies outperform the others much more clearly (approximately a rank-

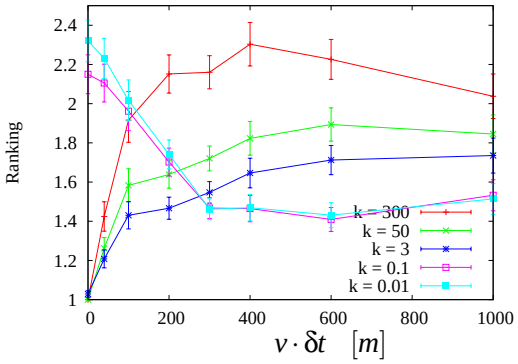


Figure 4.9: *Traversed path length with fixed sensors and mobile data sinks.*

ing of 1.4 for $k = 0.1$ and 0.01). Note that for this scenario, the ranking of a pure proximity-based strategy with $k = 300$ is much worse compared to combined strategies with $k = 50$ and $k = 3$.

4.5 Conclusion and Discussion

In this chapter, we have examined existing anycast routing strategies and introduced a new family of anycast routing schemes: density-based routing. We showed how to use field-based routing to model both existing anycast routing schemes as well as the density-based ones. We categorize the routing strategies into four types: (I) proximity-based routing; (II) proximity-based routing considering density; (III) routing as the tradeoff between proximity and density; and (IV) pure density-based routing.

Our results show that our routing model is of particular interest in dynamic networks. In these networks, a density-based algorithm achieves significant performance improvements compared to pure proximity-based approaches. We also evaluated our routing strategy in a given sensor scenario. We have also

shown that in the sensor networking environments we studied, density-based routing clearly outperforms standard proximity-based routing approaches. We even show that in highly dynamic networks, a density-based routing strategy produces shorter path lengths than pure proximity-based routing schemes.

Two open issues remain unaddressed at this point: (i) how to establish potential fields in the network in a distributed way, and (ii) how to set the value of k. Issue (i) is addressed in the next chapter when we introduce a protocol to establish potential fields. The second issue remains an open research question which we do not further pursue in this thesis. We only outline possible directions how to address this issue.

A network designer or operator could set k to best fit the expected network characteristics. However such an approach is not flexible and will not be able to adapt rapidly to changing conditions. On the other hand, since every node calculates its potential value independently of other nodes, an interesting alternative would be to let the sender decide with which value of k its packets should be routed. In that way, the sender or the sending application controls the routing strategy for its packets by assigning the value of k set in the packet header. Note that this mechanism offers, to some extent, source routing-like control without the overhead of route determination in advance.

Another alternative mechanism would be to adapt the value of k in a packet as it propagates in the network. For example, a packet is originally sent with a low value of k at the sender. As the packet traverses the network the value of k increases at each hop. This implementation dynamically changes the routing strategy as a packet is routed. At the beginning, the packet is routed to the network region with the most anycast group members. Later, packet routing converges to shortest path routing and the packet "selects" a unique destination (the closest group member). In a first step however, it would be necessary to investigate if routing still converges under these conditions.

Chapter 5

Service Discovery

In this chapter, we propose a service discovery protocol for mobile ad hoc networks that is based on field-based routing. The protocol defines service instances of the same type as point charges. These point charges create a potential field which is used by clients to find them. This chapter provides much more details about protocol issues compared to the previous chapters which were focussed on the concept and the properties of field-based routing. In particular, we no longer explore the density-proximity tradeoff by using different values of k as we did in the previous chapter. Motivated by the previous findings, we fix $k = 1$ for calculating the potential fields, resulting in a reasonable tradeoff between density and proximity.

The chapter is structured as follows. We first state the design goals of our service discovery protocol. Then, we describe the environment for which our protocol is designed and what assumptions we make regarding the underlying network technology. Afterwards, the protocol is specified and evaluated with simulations using GloMoSim, a generic wireless network simulator. Finally, we compare our protocol with other service discovery schemes.

5.1 Design Goals

This subsection specifically lays out the design goals of our service discovery protocol and briefly comments on how well it meets those goals.

- *Decentralized:* Traditional service discovery schemes as proposed for the Internet typically rely on central servers. However, in ad hoc network, we cannot assume that such servers will be available. Thus, we opt for a decentralized design without central units.
- *Robustness faced to mobility:* The degree of mobility in the network is expected to be high. Mobile nodes must be supported without degrading system performance.
- *Low overhead:* Communication resources in mobile ad hoc networks are scarce. Our protocol is thus optimized for producing low control overhead.
- *Support for multiple service instances:* The protocol is designed to support multiple service instances of the same type. The protocol is able to provide the "best instance" depending on the context of the requester.
- *Support for location-based services:* Locality plays an important role in ad hoc networks. Close service instances are favored over distant ones.
- *Routing protocol independence:* The service discovery protocol should be independent of any underlying routing protocol.
- *Determines routes to discovered services:* In addition to discovering the service, the route to the service should also be discovered. This allows for actual service invocation without relying on an underlying routing protocol.

The careful reader has certainly realized that we do not look at service discovery in the traditional sense. Service discovery in IP networks has been thought of finding the IP address of a service. Finding a route to the service is handled separately by the underlying IP routing substrate. In our work, we consider finding a route to the service as part of the discovery process. The main motivation for coupling these two tasks together is to reduce the overhead of two separate protocols since resources in ad hoc networks are scarce. However, our approach is not excluding the idea of having a separate underlying routing protocol. Our service discovery protocol could as well be use in combination of a routing protocol such as AODV or DSR.

5.2 Network Assumptions

Our discovery mechanism is entirely based on local communication and does not rely on any underlying routing protocol. The only communication service we require is at the MAC layer. We assume that it is possible to send a packet to one direct neighbor (unicast) and all neighbors (broadcast). This is for example the typical MAC service we have with IEEE 802.11 WLAN [**?**]. We further assume that all nodes in the network might be mobile and that all wireless links are bi-directional, i.e., if node s is able to transmit to node r, then node r can also transmit to node s. Although this assumption might not necessarily be true in an heterogeneous environment with different radio devices sending at various power levels, it is mainly the case for homogeneous environments with identical radio devices. Also, an implicit assumption we make is that the nodes in the network are willing to cooperate and relay packets for each other.

5.3 Discovery Protocol

In this subsection, we describe the design of our service discovery protocol.

5.3.1 General Overview

Our protocol is proactive and based on the soft state principle because we consider it unrealistic to expect service instances to de-register their profiles in a wireless ad hoc network. Service instances periodically advertise the service types they offer. These advertisements are flooded through the network within a limited scope. Each node temporarily stores recently received advertisements and calculates locally its potential value for each service type based on the information inside the advertisements. After a timeout, advertisements simply expire if they are not updated. In addition, neighboring nodes periodically exchange their local potential values for all service types. This information is needed to determine the steepest gradient.

When a client searches for a service, it creates a *service query* message. This query message contains the service type of the desired service. We assume that clients and services share a common ontology to express the service types. The sender and the intermediate nodes relay this query along the steepest ascent according to their potential value and the value of their neighbors. If a local maximum is detected, a query is forwarded to the closest service instance. The closest service instance is towards the neighbor which relayed the advertisement with the smallest distance.

When a service instance receives a query message with the service type it provides, it replies to the client with a *query reply* message. The discovery process is terminated as soon as the client receives the query reply message.

5.3.2 Protocol Messages

Four different message types are required to 1) advertise service profiles, to 2) exchange potential information between neighbors, to 3) send service queries, and to 4) reply to those queries.

Service Advertisements

Service instances periodically advertise the service they offer. A service advertisement contains the following items:

- *Service Type:* This item defines the type of service (e.g. *Internet gateway, printer, etc.*).
- *CoS:* The capacity of service which is analogous to the charge Q. Therefore, all CoS values are positive. In practice, a common quantification guideline for the CoS will be required per service type. For example, we can define the Internet gateway service capacity as follows. A service with a $100KBit/s$ connection has a CoS of 2 and a Internet gateway service with $10MBit/s$ a CoS of 15. It is also possible to adjust the CoS value depending on the momentary load of a service. A 10 MBit/s gateway could start decreasing its CoS as soon as its traffic load increases.
- *Hop Count:* The hop count field is initially set to zero by the service. It is incremented by one at each hop when forwarded. Thus, it is the distance from the receiving node to the service (in hops), as used in Equation (3.3) to calculate the potential of nodes.
- *Service ID:* An identifier which uniquely identifies the service instance. This ID is generated locally. Different mechanisms exist to achieve global uniqueness using for example part of the MAC address [?] or a time value [?].

- *Sequence Number:* A number which is incremented each time the service instance re-advertises the service it provides. This item is required to detect if the advertisement is new, obsolete, or is a duplicate which has been delivered over an alternate path.

- *Maximum Advertisement Lifetime:* A value set by the service which specifies when the advertisement expires. When an advertisement expires, its contribution to the potential must be removed.

- *TTL:* This value is set by the service provider to limit the flooding scope of an advertisement.

The way a node handles an incoming advertisement is pictured in Figure 5.1. This algorithm is required to determine if an advertisement is new, an update, or obsolete and consequently, adjust the potential value of the receiving node. When a node receives an advertisement, it first checks the *Service ID* to determine if a previous advertisement from the same originator has been received. If not, the advertisement is coming from a new service instance. In this case, the potential value for the advertised *Service Type* is created, the advertisement is stored, and then forwarded as a broadcast message to all neighbors. However, if a stored advertisement exists for that *Service ID*, the node must further look at the *sequence number* included in the advertisement. If the sequence number in the received advertisement (seq_{new}) is smaller than the sequence number from the stored advertisement (seq_{stored}), the received advertisement is obsolete and can be dropped. If seq_{new} is equal to seq_{stored}, the new advertisement is identical to the stored advertisement but must have travelled over an alternate path. The rule is to keep the advertisement which has travelled over the shortest path. Therefore, the packet is dropped if the *Hop Count* field of the new advertisement ($hops_{new}$) is larger than, or equal to, the *Hop Count* from the

stored advertisement ($hops_{stored}$). If however, $hops_{new}$ is smaller than $hops_{stored}$ or seq_{new} is larger than seq_{stored}, the contribution of the stored advertisement is subtracted, and the contribution of the new advertisement is added to the potential value. Then, the new advertisement replaces the stored advertisement and is forwarded.

Local Exchange of Potential Values

Neighbors periodically exchange information about their local potential values for the different service types. These broadcast packets have two purposes. First, these packets are used as "hello"-messages to indicate the current neighborhood nodes; thus, if a node fails to receive a packet from a neighbor for a predefined amount of time, the neighbor is assumed to be gone. Second, these packets are used to exchange local potential values of the known service types with the neighbor nodes. A node always knows the potential values of all neighbors as required to forward queries.

Service Queries

A client that searches for a service of a specific type creates a service query message. Such service query messages contain the following fields.

- *Service Type:* The service type that a client is searching for.
- *Message ID:* The message ID serves to associate a reply message from a service with a request sent by a client.
- *Requester Address:* The network address of the client.
- *Forwarding Mode:* This field is used to circumvent the inherent problem of local maxima. If a node detects that there is a local maximum in the field, the forwarding mode is switched to "proximity", and queries are forwarded to the

closest service instance instead of following the steepest gradient. The closest service instance is simply determined by comparing the *Hop Count* values from the recently received service advertisements that every node must store to calculate its own potential value.

- *TTL:* The time-to-live field is a hop count initially set by the client. It is reduced at each hop by one until it reaches zero. In that case, the query is not further forwarded. This field can be used by the client to restrict its discovery range.

- *Potential Value:* This item is used to prevent queries to loop (loops might only occur for a short amount of time until the protocol has converged). The value is set to the potential value of the node who forwards the query. If a node receives a query and detects that its own potential value is higher than, or equal to this item, it drops the query.

Query Replies

Upon reception of a service query, a service instance must reply to the client with a query reply. This query reply contains the actual network address of the service and a description field which is used to give additional information about the service to the client.

- *Service Type:* The service type of the service which replied to the query.

- *Message ID:* The ID from the query reply message is the same as the corresponding query message.

- *Service Address:* The network address of the service instance.

- *Description:* Additional information about the service. For example, the port number at which the service process is listening can be specified here.

A query reply is routed back to the client over the same path as the service query. For this purpose, intermediate nodes must store the message ID and the corresponding previous hop of service queries they forward for a small period of time.

5.3.3 Handling Node Mobility and Failures

Nodes determine connectivity by listening to the periodic potential value broadcast packets from their neighbors. If a node has not received an update packet from a neighbor for some timeout value, it assumes that the link to the neighbor is lost and removes this neighbor from its table.

As already mentioned in the previous chapters, the nodes additionally detect if neighbors have moved away or disappeared when sending unicast packets (service queries). With IEEE 802.11 [?], a node that moved away can be detected with an appropriate link layer notification (in the absence of a link layer ACK or failure to get a CTS after sending RTS). When a node forwards a service query, it tries to forward the query to the neighbor with the highest potential. However, if this neighbor has disappeared, a notification from the link layer is triggered. In such cases, the node removes the neighbor with the highest potential from its neighbor list and retransmits the query to the neighbor with the next highest potential value.

Due to mobility, it is possible that the network becomes partitioned. In this case, nodes gradually delete advertisements which timeout over time. If two network partitions merge together, services from one partition will become visible to the other partition as soon as they re-advertise their service type.

5.3.4 Reducing Flooding of Advertisements

In our protocol, service providers must broadcast advertisements periodically because this information is stored in soft state. Since these advertisements are flooded, one can argue that scalability is an issue. We therefore describe two methods specific to our approach to significantly reduce overhead traffic. Other methods to reduce flooding overhead, such as selective flooding (e.g. multipoint relaying [**?**]), could also be considered to further improve the performance. However, such schemes are not specific to our approach, but rather general improvements, and we do not discuss them in this thesis.

The first technique to reduce flooding of advertisements we propose consists of caching and aggregating advertisements before relaying them. The first time a node receives an advertisement with a *service type* it has not seen before, it adds an entry to the service table and directly forwards the advertisements to its neighbors. However, when a node receives an advertisement with a *service type* it already knows, it is not mandatory to directly forward the advertisement since a potential is already defined on the network for this service type. Hence, the node may cache the advertisement for a while. During that time, the node collects additional advertisements from other services and then forwards the collected advertisements together in one single message. With this technique, the total number of advertisement messages can significantly be reduced. Discovery performance does not significantly degrade (see Section 5.4.5) since advertisements are only delayed for existing service types. Thus, when a client sends a request during the time that an advertisement is cached at an intermediate node, the query still reaches a service.

The second overhead reduction technique consists of preventing propagation of advertisements which do not significantly alter the potential field. For example, when a received service advertisement does not modify the potential value of the receiver for

more than r percent (where r is a small value), the advertisement is not forwarded. This typically happens for advertisements from services which are far away (the distance is significantly larger than other services), or for services with a small capacity (charge).

5.4 Evaluation

In this section, we evaluate the performance of our protocol with GloMoSim, a generic wireless network simulator. Three main aspects are evaluated. We look at the performance and convergence with respect to mobility, the behavior of discovery when varying the CoS values at different service instances, and the control traffic overhead caused by the discovery protocol.

5.4.1 Simulation Model

GloMoSim [**?**] incorporates a detailed MAC and physical layer implementation. At the MAC layer, we use the available distributed coordination function (DCF) of the IEEE 802.11 [?] standard . The access scheme is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Since our evaluation focuses on service discovery, the MAC layer protocol could be replaced by other schemes without a major impact on the performance. We use a free space propagation model with a threshold cutoff for the experiments. The radio propagation range for each node is set to 250 meters and the channel capacity has a nominal bit-rate of 2 Mb/s.

In spite of the known limitations [?], we use the random waypoint model [?] as the mobility model: To make sure that the node distribution is in steady state during the whole simulation time, we place the nodes according to [?] in a rectangular area. Nodes start moving with a randomly chosen speed between 1-20 m/s to a random destination. Once the destination is reached, the node

waits for a fixed, constant pause time before another random destination is chosen. Note that with the random topologies used in the simulations, we never observed local maxima in the potential distribution. Thus, all query packets are always forwarded based on the steepest ascent of the potential and not based on proximity.

5.4.2 Simulation Parameters

The simulation duration for all experiments is set to 1000 seconds. At least twenty runs are performed for each point in the graph and the results of all runs are averaged together to produce the resulting graphs. The network size is limited to 100 nodes on a rectangular (1500m x 1300m) topology.

The value of k to calculate the potential value is set to $k = 1$ for all setups. Based on the experience we gained during our simulation studies, we set the protocol parameters to values that result in the best tradeoff between protocol performance and control overhead. Specifically, we used the following settings: Service advertisements are broadcast every 5 seconds and have a lifetime of 21 seconds (somewhat more than four times the broadcast interval). If the flooding reduction technique is used, an advertisement is cached between 0 seconds and 5 seconds (depending on the arrival time) before forwarding. Neighbors periodically exchange their potential values every 5 seconds with broadcast packets. Unless specifically described, we use the above mentioned parameters.

5.4.3 Effects of Node Motion

Two key performance metrics are evaluated to assess the effects of node mobility and to show that the protocol converges when nodes are moving. The *discovery success* is the ratio of service query packets that arrive at any service instance to the total

number of query packets sent by all clients. The control traffic *overhead* is measured as the average sending rate of control traffic per node. Control overhead traffic encompasses all service advertisements and the periodic exchange of potential values between neighbors. With this definition, the control overhead traffic is purely pro-active and therefore, independent of client requests. We do not account the service query and reply messages in the control overhead which depends on the search activity of clients. These messages are not critical to the scalability of the system since they are unicast and not flooded.

The discovery success is the most important metric, as it determines if a client discovers a service or not. The discovery success is plotted in Figure 5.2 with changing pause time. For this experiment we placed 5, 10, and 15 service instances of the same type on different nodes randomly. Ten clients are constantly sending service request packets at a rate of four packets per second. Note that this rate is much higher than we might expect in practice. We stress the network on purpose to capture the discovery performance at various moments when nodes are moving with high speed. Also, the results indicate the achievable performance when the protocol for transferring data to services as we do in the next chapter. We conclude that for higher pause times (low mobility), the discovery success is almost perfect ($> 99\%$). For lower pause times (high mobility), the performance remains fairly stable above 94%. Note that the discovery success improves when the number of services instances increases because clients and services tend to get closer on average. We conclude that the protocol converges even for high node mobility.

The control overhead per node is plotted in Figure 5.3. For this experiment, we used the flooding reduction techniques as proposed in Section 5.3.4. We will examine the control overhead without these techniques later (Section 5.4.5). The overhead is almost independent of the node mobility because control traffic is pro-active. However, the control overhead traffic rate depends

linearly on the number of service instances.

5.4.4 CoS-Distance Tradeoff

One of the interesting aspects of using field-based routing for performing service discovery, is the implicit tradeoff between distance and CoS for service selection. A close service instance is discovered by a client unless a better (higher CoS) service is available. To explore this behavior, we analyze our approach with different CoS values assigned to the service instances.

Since the CoS-distance tradeoff is the same for static and dynamic networks, the experiment was conducted without node mobility, using a static network where the nodes are placed randomly in the simulation area. We divide the set of services in a simulation in two classes, the *red* services and *blue* services. Both, the red and blue services are of the same service type but have different charges (CoS). The charge value at each service is constant over the whole simulation.

The discovery success is plotted in Figure 5.4 for different charge ratios between the red and blue services. The plot compares the percentage of requests that arrived at a blue or a red service for an increasing charge ratio. A ratio of $1:1$ means that the red and blue services have the same charge, whereas a ratio of $1:5$ means that the charge of a red service is 5 times higher than the charge of a blue service. When the charge ratio is $1:1$, we see that the service discovery queries are evenly distributed to both classes. For a charge ratio of $1:2$, 57% of client queries during the simulation arrive at red services and 43% of the queries at blue services. When further increasing the charge ratio to $1:30$, the red services are discovered 74% of the time compared to only 26% for the blue services. Further intensifying the charge ratio does not much impact the distribution any more and we therefore conclude the following. When the charge ratio is $1:1$, clients discover services which are very close independent of their CoS.

As the charge ratio increases, discovery packets start to drift in the direction of red services as the potential gradient gets steeper in that direction. Thus, a discovery packet can be forwarded to a red service even if a blue service is closer. Note that the charge ratio does not influence query packets from clients which are one hop away from a service instance because the potential value at a service is very large (infinite) and therefore, this service instance is always chosen as a next hop.

Another aspect of the charge ratio is reflected in the range or scope of the potential field. Specifically, we examine the reach (measured as the number of hops between the client and the discovered service) of high capacity services compared to low capacity services. To illustrate the influence of the charge ratio on the range, we plot the distribution of the distance between clients and discovered services for three different charge ratios in Figure 5.5 to 5.7. For each query that arrived at a service, we determined the distance from the requesting node to that service. We then plotted the share of queries that arrived from nodes with a network distance of 1 to 8 hops.

When the charge ratio is $1:1$ (Figure 5.5), the distribution for the blue and red services is essentially identical (the red and blue curves will converge with an infinite number of simulation runs). It is interesting to see the distribution behavior when increasing the charge ratio. In Figure 5.6, the distance distribution is plotted for a charge ratio of $1:5$. Blue services tend to only get discovered by close clients. This effect is even more pronounced when the charge ratio is increased to $1:30$ (see Figure 5.7). The larger the ratio, the more dominant is the high CoS service instance. We conclude that CoS is an effective mean to differentiate service instances and balance the load among them.

# services	flood	reduced
5	94.68%	94.61%
10	97.01%	96.91%
15	97.63%	97.61%

Table 5.1: *Discovery success with (reduced) and without (flood) reduction technique.*

5.4.5 Improvements by Overhead Reduction Technique

We next investigate how much control overhead is saved by the proposed flooding reduction technique which consists of aggregating service advertisements (see Section 5.3.4). To determine how effective our optimization is, we first compare the discovery success with and without optimization for 5, 10, and 15 service instances and high node mobility (pause time = 0 seconds, $v_{max} = 20m/s$). Then, we compare how much control overhead is caused with both approaches.

Table 5.1 shows the discovery success for different service numbers. In the different scenarios, we observe only a small, acceptable performance loss of $\leq 0.1\%$.

We now evaluate the efficiency of our control overhead reduction technique. In Figure 5.8, we see that up to 54% of the total traffic load per node was reduced per node. In addition to the average sending rate of control traffic per node, we also measured the average number of control packets sent per second at each node. The result is shown in Figure 5.9. The number of control packets that were sent are reduced approximately by a factor of 5 (for 5 services) to a factor of 14 (for 15 services). We further observe that, using the flooding reduction technique, the number of control packets does not increase with the number of services. This is because advertisement packets from different service instances can be combined into a single advertisement packet.

5.5 Comparison with Other Service Discovery Schemes

In this section, we compare the performance of field-based service discovery with two other schemes. A fair comparison is only possible with proactive and network-based schemes where service instances actively advertise themselves and establish routing state in the network. Reactive discovery schemes, where routes to services are established on demand only when services are requested by clients, are hence not considered. The basic question of whether a proactive scheme is more efficient than a reactive one, or vice versa, is outside the scope of our evaluation. In general, there is no simple answer to this question and the answer strongly depends on factors such as the number of services, the number of clients, the requesting rate of clients, and the network dynamics.

5.5.1 Service Discovery Schemes

The most popular method for network-layer service discovery in datagram networks is the use of anycast routing [?] [**?**]. We distinguish two mechanisms, anycast-1 and anycast-N, to implement anycast using *service advertisements* messages introduced previously in this chapter, and compare these two schemes with the performance of field-based service discovery.

Service Discovery with Anycast-1

In the anycast-1 service discovery scheme, every node receives the service advertisements from the different service instances and stores only *one single* entry in its routing table, the one towards the neighbor which sent the advertisement with the smallest *hop count* value. Therefore, a query is always sent to the neighbor that is the closest to any service instance. The major drawback

of this simple anycast implementation is its lack of robustness. Due to its single entry per service, it fails to deliver a query when any of the links on the path to the service becomes unavailable.

Service Discovery with Anycast-N

The anycast-N service discovery scheme is a more sophisticated variant of anycast-1. Each node maintains a list of the last received advertisements and from which neighbors these advertisements originated. Therefore, if N is the average number of neighbors per node, on average each node stores N entries in its routing table per service type (which is why we call it anycast-N). As with anycast-1, at each hop, the query is always forwarded to the neighbor closest to any service instance. However, if due to a link failure this neighbor is no longer reachable (detected, for example, in the absence of a link layer acknowledgement), the node determines another neighbor with the next closest service instance and sends the query to this neighbor. To avoid loops with this anycast implementation, one must make sure that the hop count (the smallest distance from a node to any service instance) is reduced at each hop, as a query is propagating through the network.

Field-based Service Discovery

Field-based service discovery is the scheme we propose in this chapter. Since we assume that all service instances have the same capacity, the same charge Q is assigned to every service instance. In this scheme, queries are not necessarily routed towards the closest service instance as with anycast-1 and anycast-N; by using a value of $k = 1$, the service concentration in a specific direction is also considered so that more distance service instances can be preferred to close ones.

5.5.2 Robustness

The robustness of the different schemes is assessed with simulations by determining the discovery success for different degrees of node mobility. We model the network as a graph. A graph is generated by placing nodes randomly on a two-dimensional plane and assigning an edge between two nodes if their geometric distance is smaller than a value which models the wireless range of the node's radio device. For simplicity, node mobility is modeled with a link connectivity model [**?**] which consists of removing existing edges of the graph according to a uniform probability distribution. Note that we observe the same trend by accounting for the link failures caused by a mobility model such as for example the random waypoint mobility model. Clients and services are assigned randomly to the nodes of the graph.

The discovery success for graphs of 400 nodes with 5 service instances is plotted in Figure 5.10 (a) and for graphs of 600 nodes with 10 service instances in Figure 5.10 (b). The average client-service distance in both experiments is around four hops. For each point in the plots, at least 4000 different topologies were used. The horizontal axis of these two plots shows the fraction of removed edges per service advertisement interval (which is the period between two consecutive advertisement messages of the same service instance). The discovery success plotted on the vertical axis is the ratio of successful discoveries to the total number of discovery attempts of all clients.

As expected, all three schemes perform equally well in a static network without mobility (no links are removed). However, the performance of anycast-1 degrades rapidly when links disappear. The poor performance is caused by the limited level of redundancy at each node. Since nodes store only one entry for the shortest path towards the closest service, a query can no longer be delivered when any link on this path becomes unavailable. The stability is significantly increased with anycast-N. With anycast-

N, when a link becomes unavailable on the shortest path, a query can still be routed over an alternative path if one exists. Note that as with anycast-1, each node always tries to forward a query on the shortest path to the closest service. This is the main difference to our field-based service discovery scheme. With field-based service discovery, queries are routed towards the steepest gradient of a field. Therefore, queries with $k = 1$ as we set in this chapter are not necessarily routed towards the closest service. Network regions with lots of services are favored over a close by, isolated service. This strategy clearly improves the robustness of the system as we can see in the plots from Figure 5.10. Note also that the robustness of field-based service discovery is already high with small number of service instances.

It is remarkable that the stability of field-based service discovery clearly outperforms the other two schemes even when services are placed randomly. The benefits of a field-based service discovery scheme are however even more pronounced when the services are arranged according to a certain structure. To outline this behavior, we conducted an experiment with specific topologies. The results from this experiment should provide an intuition for topologies where field-based service discovery is particularly favorable. In this experiment, the services are placed according to Figure 5.11 (a). Nine services (marked with S) are placed randomly on the left part, and only one service on the right half part of the simulated network. The client (marked with C) is placed in the middle of the simulation area. The result of this experiment is shown in Figure 5.11 (b). The stability of the field-based service discovery scheme is even more pronounced than in scenarios when services are placed randomly. The reason is the following. Since nine services are located on the left side of the client, the potential is much higher in this direction than towards the other side. Therefore, queries using field-based service discovery are always routed towards the left side. However, with anycast-1 and anycast-N, if the service on the right half side is closer than any

other service, a query from the client is routed towards this service. If links on the path towards the isolated service instance become unavailable, it is then not possible to find an alternative service instance in the neighborhood.

5.5.3 Service and Route Optimality

With field-based service discovery, when using $k = 1$, there is no guarantee that a query is delivered to the closest service instance. There is also no guarantee that a packet is delivered to a service instance over the shortest path. However, when all service instances have the same capacity, in this specific case, it is desirable that a client discovers the closest service. Therefore, we assess if the client-service distance achieved by field-based service discovery using $k = 1$ is not significantly higher than the *optimal distance*: the length of the shortest path to the closest service instance. We therefore compare the average path length a query travels until it reaches a service with anycast-1, anycast-N, and field-based service discovery. Anycast-1 and anycast-N are, according to our definition, optimal since they compute the shortest path to the closest service instance. We therefore use these two schemes as benchmark for our measurements.

For the performed experiment, we used static network topologies with uniformly distributed nodes. In order to get similar client-service distances when varying the number of service instances, we adjusted the network size in order to keep an average client-service distance between 3 and 4 hops with anycast-1/anycast-N. Then, we used the same network topologies to measure the client-service distance with field-based service discovery. The results are promising. As we can see in Figure 5.12, the average client-service distance of field-based service discovery is close to the optimal value. For 15 service instances, the average distance is only increased by 5.3% compared to anycast-1/anycast-N. For one service instance, the distance is optimal as proved in

Section 3.2.1 of this paper. We conclude that the increased client-service distance with field-based service discovery is very small, and thus tolerable.

5.6 Discussion and Conclusion

This chapter defines a new service discovery protocol based on field-based routing for mobile ad hoc networks. Our approach combines the task of routing and service discovery into one protocol. That is, query packets are directly forwarded to the matching service. There is no intermediate lookup in this scheme.

One obvious advantage of this scheme is that we produce less overhead than a scheme which decouples both tasks. A second advantage is that by applying the concept of field-based routing, we are able to discover services which are in direct proximity, and it is also easily possible to account for services with different capacities. Furthermore, our protocol is inherently robust to link failures and mobility as we inherit the density delivery characteristics of field-based routing.

The natural continuation is to extend the discovery protocol to provide a full communication service between the client and the service. The challenge at the routing layer is to make sure that consecutive packets reach the same service instance when the service is statefull. For statless services, the discovery protocol is sufficient. In the next chapter, we address the issue of how the full communication architecture must be designed for this purpose.

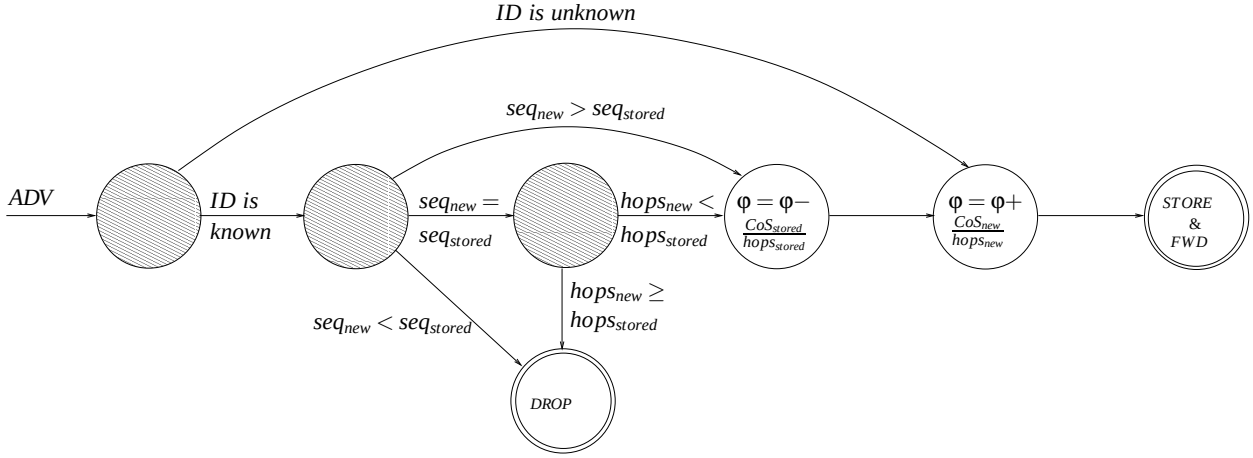


Figure 5.1: *Advertisement handling.*

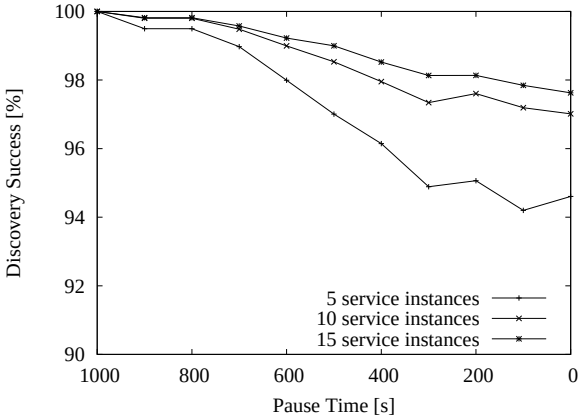


Figure 5.2: *Discovery success for different pause times.*

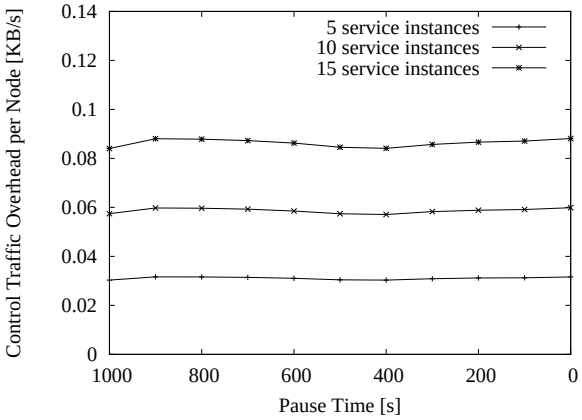


Figure 5.3: *Control traffic overhead per node for different pause times using the flooding reduction technique.*

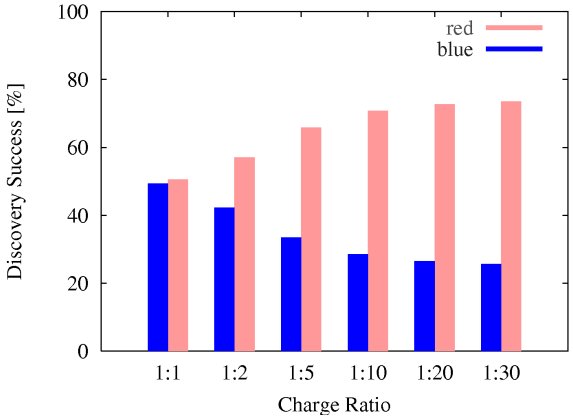


Figure 5.4: *Effects of different CoS values.*

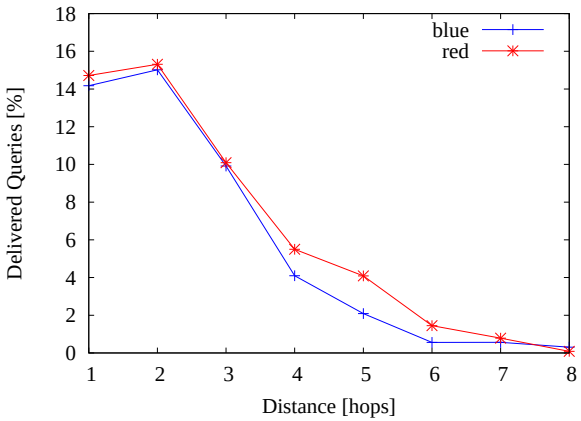


Figure 5.5: *Charge ratio 1:1 - The distribution of the distance between client and discovered service.*

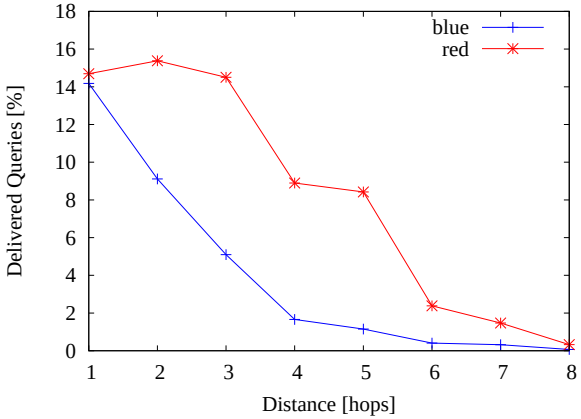


Figure 5.6: *Charge ratio 1:5 - The distribution of the distance between client and discovered service.*

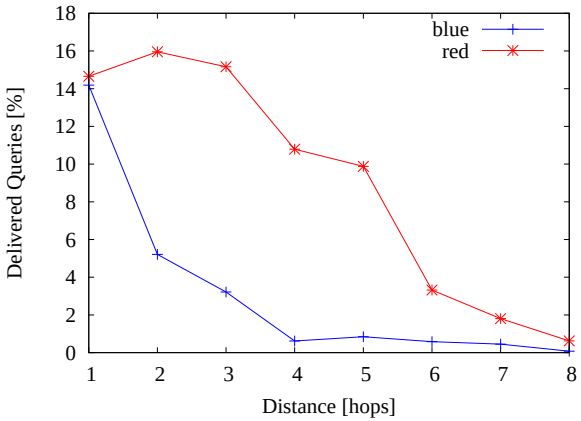


Figure 5.7: *Charge ratio 1:30 - The distribution of the distance between client and discovered service.*

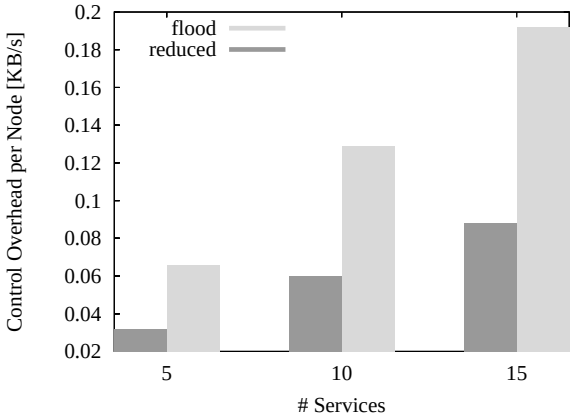


Figure 5.8: *The control overhead per node measured as the average sending rate without optimization (flood) and with optimization (reduced).*

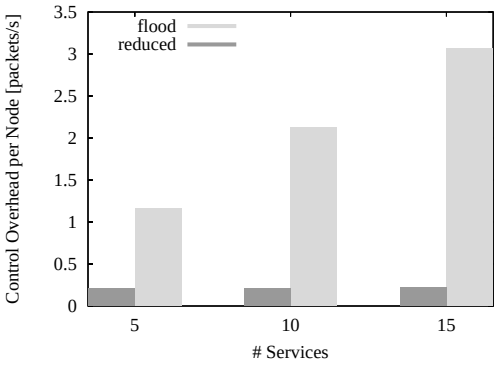
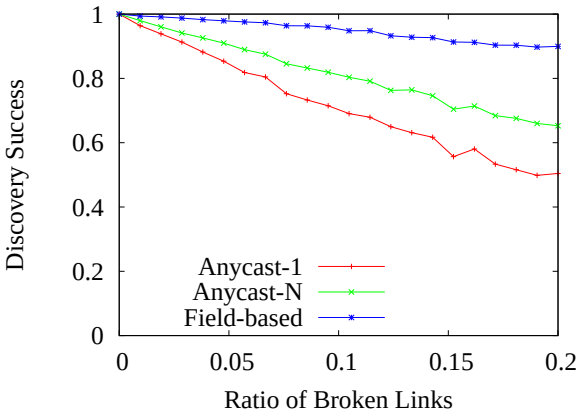
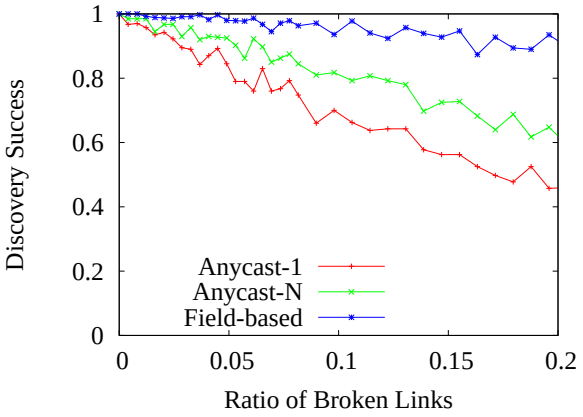


Figure 5.9: *The control overhead per node measured as the average number of control packets sent without optimization (flood) and with optimization (reduced).*

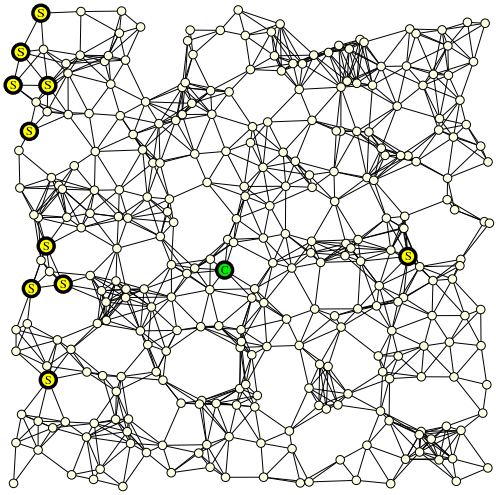


(a) 5 service instances.

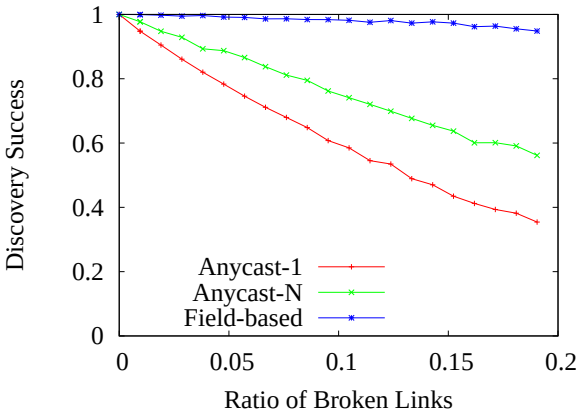


(b) 10 service instances.

Figure 5.10: *Service discovery stability with random topologies.*



(a) Structured network topology.



(b) Discovery success in structured topologies.

Figure 5.11: *Service discovery stability with structured service topology.*

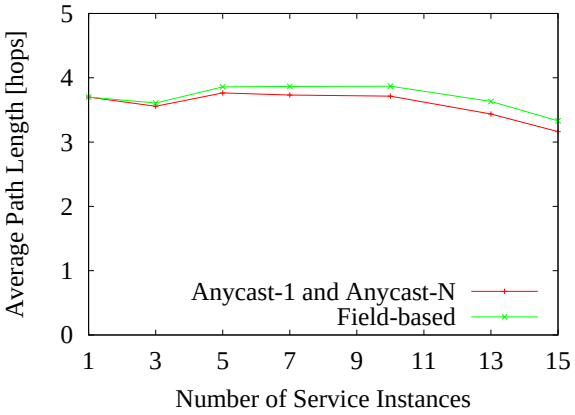


Figure 5.12: *Client-service distance.*

Chapter 6

Autonomic Communication with Field-based Routing

In this chapter, we propose a network architecture that relies on field-based routing. The chapter is organized as follows. We first justify the need for a new network architecture in mobile and ad hoc networks as an alternative to IP the protocol suite. Then, we present the design and implementation of our architecture. We conclude by highlighting the benefits of our communication architecture compared to IP.

6.1 Motivation

Many research groups [**?**, **?**, ?, ?, ?, ?, ?] designed protocols for MANETs that rely on the Internet protocol (IP). We believe that IP, including IP-based routing, does not provide the appropriate set of abstractions for wireless ad hoc networks. In the following, we describe reasons why we believe this is so.

- *Host- and Location-based Identifiers:*

  IP provides only identifiers (IP addresses) for hosts. Although IP networks are most frequently used to gain access to services that are remotely invoked by clients, there is no name space to directly and persistently identify those services. Instead, services instances and clients are named relative to the hosts on which they reside. E.g., A service running on host with IP address *ip-address*, and listening on port 80, is referred to as *ip-address*:80.

  A second limitation is that host identifiers (IP addresses) have inherent topological meanings and depend on the location of the hosts. This design choice results from the fact that the Internet architecture was originally designed for fixed hosts. However, this design choice is a severe limitation for mobile networks. When hosts are mobile and move over subnet boundaries, they are forced to change their IP address and thus their identity! This is a problem for the applications that rely on this identity to communicate, and also for the management of addresses/identities which becomes abundant when the hosts are moving a lot.

- *Binding Semantics:*

  When clients request a service, they care only about the service they request; the particular instance servicing a request is irrelevant. However, the IP architecture obliges clients to resolve a service type down to a host (the IP datagram structure requires a sender/client to specify the IP address of the receiver/service), thereby forcing a client to bind to a specific service instance. Therefore, we argue that the IP protocol suite imposes unnecessary restrictions to applications.

- *Layering:*

The design of the Internet architecture is based on a hierarchy of independent protocol layers (see the TCP/IP reference model [**?**]). In this layering, the process of finding a service and finding a route to this service is handled by two different layers (routing is handled at the network layer and service discovery at the application layer above the transport layer). However, in ad hoc networks, keeping service discovery and service location separate is inefficient: the network is confronted with the overhead of two protocols instead of one.

- *Robustness:*

  IP routing was designed to run on "reliable" link layers. Packet losses on wired networks are rather the exception. As a result, the routing techniques used (link state routing, distance vector routing, etc.) aim at strictly optimizing the delivery costs (e.g., the number of hops). However, wireless and in particular ad hoc networks have unreliable link layers (packet losses of more than 10% are the common case). We argue that in such conditions, it is necessary to consider routing techniques which also aim at optimizing the robustness (such as for example field-based routing).

- *Security:*

  The Internet architecture has shown to be extremely vulnerable to certain type of attacks. For example, (distributed) denial of service (DoS) attacks have recently caused large economic damage [**?**]. In a DoS attack, one or more malicious hosts send a large amount of unwanted traffic to a victim, unable to process all incoming traffic. As a side-effect, the victim is no longer able to handle its desired traffic. One direct weakness of IP with this regard is that the architecture's goal is to provide end-to-end connectivity to any hosts in the network. However, what is desired

in most of the cases is the ability to route service requests to a server for that service, and to route responses back to the client and nothing else. In this case, the architecture provides functionality which is not necessary for the primer usage (accessing a service) of the architecture but used for malicious purposes. Indeed, global client reachability is typically not required and attacks to clients could be mitigated by limiting reachability (routing state) to a local scope between the client and requested service.

6.2 Design

Motivated by the current limitations of the IP protocol suite, we design a new network architecture for mobile ad hoc networks. The design of this architecture is presented in this section.

6.2.1 Architecture Overview

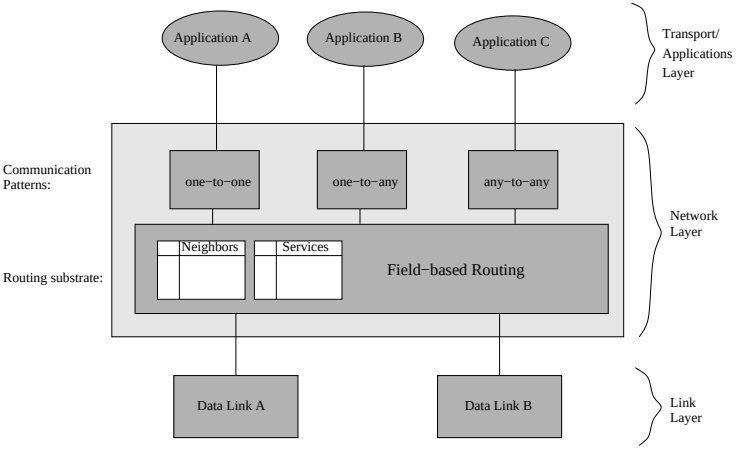


Figure 6.1: *Architecture overview.*

Our architecture consists of a network layer with, in addition to routing functionality, support for service discovery and binding. This is fundamentally different from the TCP/IP architecture which handles service discovery and binding functionality above the transport layer. A schematic view of the architecture is shown in Figure 6.1. The network layer is divided into a routing substrate and individual communication patterns. The routing substrate is a best-effort anycast delivery layer and the communication patterns on top of it define the binding semantics between application end-points. We define three typical client-service patterns for communication between (i) two fixed end-points (one-to-one), between (ii) an end-point and a group of potential receivers (one-to-any), and between (iii) any sender and any receiver (any-to-any). The pattern chosen depends on the service or the application requirements. For example, when the remote service is a printer, a client uses the one-to-one pattern to ensure that all the packets of a single printing job is delivered to the same service instance. The one-to-any pattern is used by stateless applications such as for example an Internet gateway service or query-reply type of services. For such services, consecutive packets of the same session can be delivered to a different service instance. The any-to-any communication pattern performs no binding at all. This type of communication is typical for sensor type of communication or alarm services where a message is sent over a single data packet to any data sink without the need for replying data back to the sender.

The network layer has one *global* identifier space which is used to route packets at the routing substrate. An identifier in this space refers to a group of entities such as for example services of the same type. Additional *local* identifier spaces may be defined by individual communication patterns. For example, the one-to-any pattern needs an additional identifier space to uniquely identify entities that expect reply packets back. However, for the any-to-any pattern, the global name space is sufficient.

The network layer has a well-defined service abstraction towards the lower layer so that multiple data link technologies can be used. We next discuss the individual components of the network layer in more details.

6.2.2 Routing Substrate

The routing substrate provides a best-effort anycast delivery service. Note that we implement this layer with the field-based service discovery protocol introduced in the previous chapter and refer to it for the protocol details. Our main motivation for this is to profit from the benefits of this scheme in mobile networks. In principle, other implementations of an anycast routing substrate could be used as well.

For the sake of generality, and since we do not expect humans to be confronted directly with the global name space of the routing substrate, we adopt for a name space which is flat and semantic-free instead of using human-readable canonical or even hierarchical-structured names. The main driver for this design choice is that flat semantic-free names do not put any restrictions on the objects/services they refer to, thus being more flexible, and at the same time we want to achieve name persistence by not imposing location-dependency which makes movement hard as we argued against IP addresses. Identifiers in this name space are referred to as STID which is an acronym for *service type identifiers*.

A legitimate question is how humans express the type of service their are looking for, since STIDs are free of any semantics. Imagine a user that is looking for a printer, how does he know the STID of the service type "printer". There are basically two possible approaches to address this issue. The first idea would be to implement a lookup service in the network which maps search queries or human-readable names to STIDs. A second approach would be that services advertise themselves by specifying their

STID in conjunction with a description of the service provided. These advertisements could then be used by humans to retrieve information of the surrounding services including their STIDs. Both approaches have pros and cons. In what follows, we assume that humans and the applications under their control have access to such an auxiliary service and leave the implementation issues of such a service to future work.

6.2.3 Communication Patterns

Communication patterns define the binding semantics between clients and services in our architecture. We introduce three typical patterns in this thesis, but in principle, other patterns could also be defined as well. In addition to the binding semantics, each pattern defines a set of protocols, and if necessary, additional local name spaces to achieve the desired binding.

In the following, we describe the any-to-any, the one-to-any, and the one-to-one patterns. In all three patterns, a sender starts to communicate by sending a data packet addressed to the global STID of the desired service type. This packet is delivered to any entity that has this STID. The patterns only differ in the remaining part of the communication.

The Any-to-any Pattern

The any-to-any pattern is the simplest form of communication we might implement on top of the field-based routing substrate. It does not impose any binding between a client and a service instance in either direction. This pattern does not require additional mechanisms. A client addresses its packets with the desired STID and packets are delivered to any entity that provides that service. Note that it is not possible for the receiver of the packet to reply back to the originator of the packet. Figure 6.2 figures a schematic view of this pattern. The first packet is delivered to

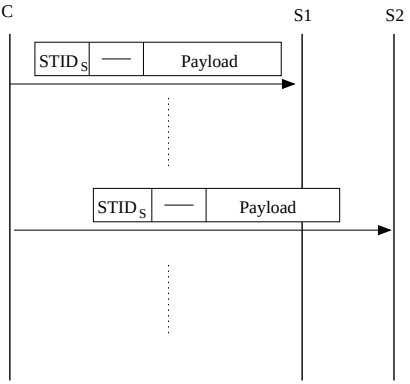


Figure 6.2: *Any-to-any pattern.*

*S*1. A later packet to the same STID can possibly be delivered to another entity, in this case *S*2.

Note that this type of communication is unreliable by design since it is not possible to send a feedback (positive or negative) from the service back to the client which would be necessary to guarantee successful delivery. Such a pattern should only be used by application which can cope with such a unreliability.

The One-to-any Pattern

Applications using this pattern do not care about the instance of the service provider but require reply data back from the service. To be able to send reply packets back to clients, this pattern defines a new name space of unique identifiers (UIDs) for the clients and a protocol to setup up a reverse path from the service to the client while the client sends its first packet.

The protocol is shown in Figure 6.3 and works as follows. Before the communication starts, a UID is created at the client node. This UID serves to identify the client at the service and at the intermediate nodes which are on the reverse path to the

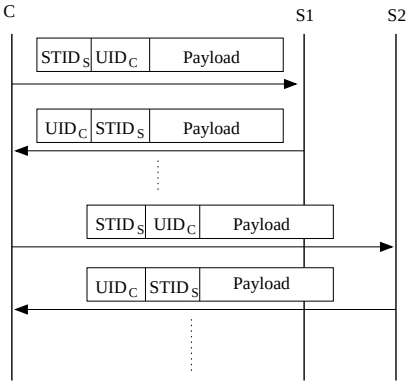


Figure 6.3: *One-to-any pattern.*

client. The UID is appended to all packets addressed to the service that the client sends. As those packets travel along the path to a service, the intermediate nodes establish a reverse path for possible reply packets from the service. When a service receives such a packet, it replies by using the UID included in the packet header and the reply packet can now be routed back along the potential field for this UID.

Note that UIDs have only local significance, namely at the client, the receiving service, and the intermediate nodes between them. We propose to use semantic-free identifiers for UIDs, similar to STIDs. To make sure that duplicated UIDs are not generated on different hosts, UIDs are random numbers generated from a large set (e.g., ≥ 128 bits) providing statistical guarantees of uniqueness.

This pattern can be used by query-reply type of applications or applications which require reliable data transfer: the reverse path is then used to send acknowledgements packets back from the receiver to the sender.

The One-to-one Pattern

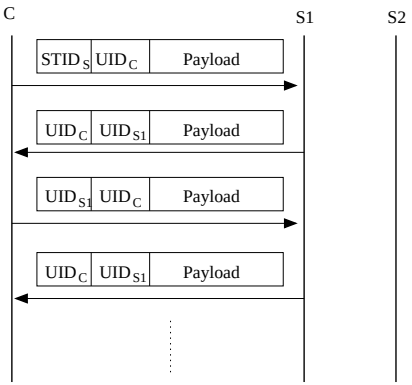


Figure 6.4: *One-to-one pattern.*

The one-to-one pattern is used by applications that require binding between the client and the service provider. This pattern introduces a similar protocol as for the one-to-any pattern and a name space of unique identifiers (UIDs) for clients, but now also for services. This name space could be shared with the previous pattern as it has the same properties.

The details of the protocol is sketched in Figure 6.4. The client creates a UID (UID_C) and appends it to its first packet as with the one-to-any pattern. Again, the intermediate nodes store the UID of client and establish a reverse path to it. An alternative would be that this packet creates a potential field for the UID of the client towards the receiver. This approach would be more robust since it could handle link failures along the reverse path by routing along the established potential field. In this thesis, we assume that the lifetime of the communication is short, so that we do not have to account for these type of failures. However, for longer communication lifetimes, it would be necessary to investigate the latter case which we refer to future

work.

When the service receives the first packet from the client, it also creates a UID (UID_{S1}), and appends it to its the reply packet. This reply packet is also used at the intermediate nodes to establish a reverse for UID_{S1}. From now on both, the client and the service, communicate using their respective UIDs. STIDs are no longer used. Note that if the service already has a UID (because it created one earlier to communicate with another client), it is not necessary to generate a new UID, but the same UID should be used to communicate with both clients.

6.3 Proof-of-Concept Implementation

To show the feasibility of our approach, we have implemented our architecture and made functionality tests with a IEEE 802.11b testbed. Our implementation is written in ANSI C, to take the advantage of its cross-platform portability and its fast execution compared to other programming languages. Our user-level implementation is divided into several threads and uses raw sockets to access the network stack directly, without having to pass through the IP stack.

In this section, we start by describing the defined packets formats and explain how we have implemented and tested our implementation.

6.3.1 Packet Formats

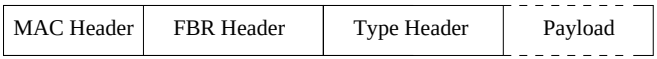


Figure 6.5: *FBR header.*

Data packets are composed of a sequence of headers as shown in Figure 6.5. The first header is the MAC header of the under-

lying link layer technology (IEEE 802.11b in our case). Directly on top of it, there is a generic network protocol header (FBR header). The last header before the data payload is specific to the message type. We differentiate four different message types:

- `MSG_SERVICE_ADV`: Service advertisement message. This type is defined in Chapter 5.
- `MSG_NEIGHBOR_EXCHANGE`: Neighbor exchange packet. This type is also defined in Chapter 5.
- `MSG_SERVICE_QUERY`: Data message destinate to services via their STID.
- `MSG_REPLY`: Data message including the UID of the destination node.

In the following, we introduce the header format of the generic FBR header, the `MSG_SERVICE_QUERY`, and the `MSG_REPLY` headers. The packet types for `MSG_SERVICE_ADV` and `MSG_NEIGHBOR_EXCHANGE` are almost identical to those introduced in Chapter 5 and thus not further discussed in this chapter.

FBR header

The FBR header (see Figure 6.6 for a layout) is a generic header for all packets. It contains the following fields:

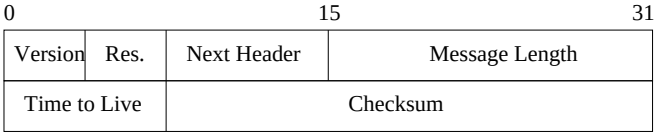


Figure 6.6: *FBR header.*

- *Version (4 bits):* Version of the protocol, currently 1.

- *Reserved (4 bits):* Reserved for further use.
- *Next Header (8 bits):* Defines the subsequent message header type. Currently, four different packet types exist.
  - Type 1: `MSG_SERVICE_ADV`
  - Type 2: `MSG_NEIGHBOR_EXCHANGE`
  - Type 3: `MSG_SERVICE_QUERY`
  - Type 4: `MSG_REPLY`
- *Message Length (16 bits):* Length of the payload (including the subsequent headers but excluding the MAC and this header).
- *Time to Live (8 bits):* This field is decremented at each hop. When it reaches zero, the packet is dropped. This field is to assure that packets do not loop forever in the network.
- *Checksum (16 bits):* The checksum value serves to check if the header was transmitted correctly.

Service Query Header

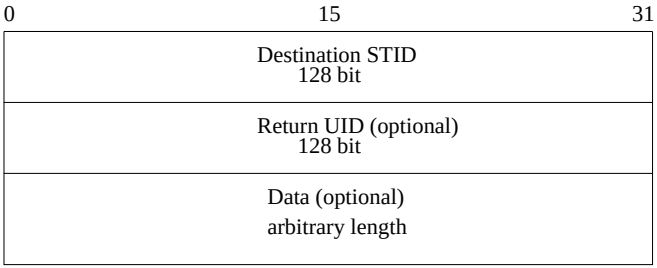


Figure 6.7: *MSG_SERVICE_QUERY header.*

Service query packets are packets that are routed based on a STID to any service instance. These packets are structured as

shown in Figure 6.7. The header fields have the following meaning:

- *Destination STID (128 bits):* The STID of the destination for this packet.
- *Return UID (128 bits):* The UID of the sender. This value is required to send packets back to the sender. This field is optional and can be discarded by setting the UID value to a sequence of 128 x 0 bits if the sender does not expect data back. In that case, the value has no meaning.

Message Reply Header

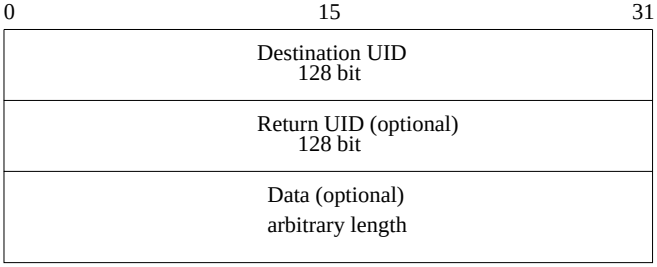


Figure 6.8: *MSG_REPLY header.*

Reply messages (see Figure 6.8) are packets sent to a specific entity. These messages always follow after a query packet. The destination UID of reply packets is namely learned from those.

- *Destination UID (128 bits):* The UID of the destination entity for this packet.
- *Return UID (128 bits):* The UID of the sender. This value is required to send packets back to the sender. This field is optional and can be discarded by setting the UID value to 0 if the sender does not expect data back. In that case, the value has no meaning.

6.3.2 Node Implementation

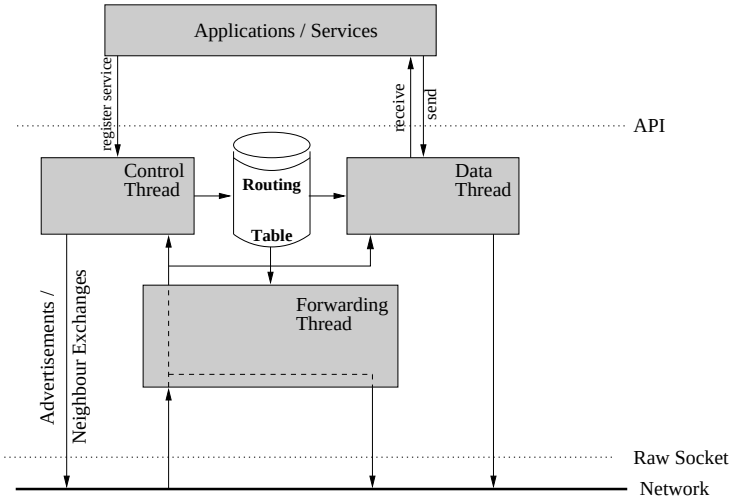


Figure 6.9: *Node implementation.*

Our node implementation is sketched in Figure 6.9. The tasks of handling control messages, handling data packets, and forwarding packets are divided into three separate threads. The *control thread* feeds the routing table with the information it collects from received service advertisements and neighbor exchange messages. Both, the *data* and the *forwarding thread* rely on the routing table to forward packets to the correct neighbor. Our implementation accesses the network layer directly via raw sockets to bypass the TCP/UDP and IP stack.

Applications or services communicate with the system via an application program interface (API). The API provides two control functions to *register* and *de-register* services. It also provides a *send* function, and a call back function to *receive* data from other nodes.

When an application invokes the *send* function, it specifies a

STID and the desired communication pattern (any-to-any, one-to-any, or one-to-one). Depending on the chosen pattern, the *data thread* handles the creation of the required UIDs and follows the appropriate protocol to achieve the desired binding. The application itself is unaware of UIDs; it relies on the underlying system for that purpose.

6.3.3 Test Network

Our test network consists of five Laptops and one desktop computer. The used laptops were IBM Thinkpads R32 having an integrated WLAN adapter based on an Intersil Prism 2.5 chipset. The desktop computer had a PCI WLAN adapter based on the same chipset. The computers were running Debian Linux (kernel 2.4.24 and 2.6.10) with the orinoco WLAN driver. The WLAN interfaces were turned into ad hoc mode.

The test network was used as a proof-of-concept prototype to show the feasibility of our approach. For each pattern we ran a series of three functional tests, to test if the protocols were functioning correctly when the nodes were arranged in a long chain, a node failed, and a node moved during the communication. In those scenarios, our implementation worked properly and we conclude that the design principles of our architecture can be implemented. We are aware that a larger network with more nodes and unpredictable mobility would be necessary to evaluate the scalability of the system. However, due to the unavailability of such a network, we rely on the previously shown simulation results for this.

6.4 Applications

To understand the involved steps in the communication with our architecture, we describe in this section the workflow of three applications. The first application is a notification service and uses

the any-to-any communication pattern. The second application is about providing Internet access to nodes in the ad hoc network. This application has a one-to-any communication pattern. The last application is a printing utility service. The communication pattern in this scenario is one-to-one.

6.4.1 Notification Service

The notification service is an application to send short messages. This could for example be used to send alarm messages in a fire detection sensor network or to send orders from customers to the waiters in a restaurant. The steps involved in this application are:

1. The notification receivers (e.g., the waiters) start their application which register their STID with the *register("STID")* function from the API.

2. After the registration, the *control thread* on those nodes starts to advertise the STID by sending periodic `MSG_SERVICE_ADV` messages. This establishes a potential field for this STID in the network, or adapts an existing field in case another node previously advertised the same STID.

3. A client node that wants to send a message, invokes the *send* function and specifies the destination STID as well the communication pattern (any-to-any in this case). On the applications behalf, a `MSG_SERVICE_QUERY` message is created by the network layer. The header of this packet contains the STID specified by the client application and a return UID set to zero to indicate that no return path should be established for this packet.

4. The `MSG_SERVICE_QUERY` is forwarded along the steepest gradient of the field until it reaches any node which advertised the STID. When the packet arrives at its destination, the packet payload is delivered to the application process.

6.4.2 Internet Gateway Service

The Internet gateways provide an Internet connectivity service to nodes within the wireless ad hoc network (one-to-any pattern). The goal of this application is to be able to communicate with Internet hosts from the ad hoc network. Thus, a client sends and receives IP traffic which is tunneled over the ad hoc network. The steps involved with this application are the following:

1. Nodes which have an active connection to the Internet register the corresponding STID with the *register()* method. The registration initiates periodic `MSG_SERVICE_ADV` packets which establish a potential field for this service type.

2. A client node sends its IP traffic via the *send()* function and specifies the gateway service STID, as well as one-to-any as communication pattern.

3. The *data thread* running on the client first creates a new UID.

4. The data thread then sends the client data packetized in `MSG_SERVICE_QUERY` messages with the destination STID of the gateway service and the previously created UID as return UID.

5. Intermediate nodes which forward `MSG_SERVICE_QUERY` messages store a reverse path entry for the UID in the packet header.

6. Any node with Internet connectivity which receives such a packet sends the packet's payload consisting of IP traffic to the corresponding IP destination address. IP packets coming back from the Internet, are packetized in `MSG_REPLY` messages with the destination UID of the client and forwarded back along the previously established reverse path.

The return UID in the header of the reply message is set to zero as it is unused.

6.4.3 A Printer Utility Service

We show with the printer utility service how an application with a one-to-one communication pattern is implemented.

1. Printers register their service and establish a potential field in the same way as shown in both previous applications.

2. A client sends a print job via the *send* function and specifies one-to-one as communication pattern.

3. The *data thread* on the client creates a UID.

4. The data thread on the client packetizes the print job but only sends one packet to the print utility service STID and then waits. The return UID of this packet is the UID created in the previous step.

5. The intermediate nodes which forward the packet store a reverse path entry for the client UID. At any printer which receives this packet, the *data thread* handles the packet payload to the application layer, and then creates a new UID.

6. The data thread at the printer then immediately sends back a `MSG_REPLY` to the client with the return UID that has been created in step 5. Note that this packet does not necessarily contain any data in the payload but serves to inform the client about the printer's UID.

7. This packet establishes a reverse path to the printer at the intermediate nodes. As soon as it reaches the client node, the data thread of the client node sends the remaining packets of the print job as `MSG_REPLY` messages with the UID of

the printer as destination UID. With this handshake mechanism, we make sure that all packets of the print job are delivered to the same printer.

6.5 Discussion

In this section, we compare our approach with the IP protocol suite, and show the benefits as well as the limitations of our design choices.

6.5.1 IP Addresses vs. UIDs

UIDs serve to uniquely identify clients and service instances in the network. In a sense, a UID is similar to an IP address which is also used as a unique identifier. However, UIDs are different from IP addresses in many ways:

- UIDs (as well as STIDs) use a flat semantic-free name space, whereas IP addresses are structured and have a topological meaning. Our approach allows entities to move without the need to change their identity (their IP address).

- UIDs have a local scope. They have only a meaning for the communicating entities and the surrounding nodes.

- UIDs are generated on the fly when needed. This autonomic paradigm alleviates the need of identity management for assigning unique identifiers to all entities (collisions of identical UIDs are avoided by using a large id space as mentioned previously).

- A UID is not assigned to a network interface but rather to a client or service process. Therefore, a service or a client entity running on a device with multiple interfaces does not

have to use different identities for each interface. Furthermore, when a service or client is migrated to another host, it can keep its identity.

- Finally, UIDs are only assigned to clients and services. Intermediate nodes or routers which do only forward packets in the network do not a have a UID or any network identity at all.

6.5.2 Combining Service Discovery and Routing

Pushing down the service discovery mechanisms at the network layer has two major advantages. First, it allows to design a single protocol for distributing routing and service information state, thus reducing control overhead. Second, it allows for situated communication where clients are binding to service type identifiers (STIDs), and thus always reach the closest service instance as they move or as the topology changes.

6.5.3 Routing Optimality vs. Robustness

Traditional IP routing mechanisms such as link state or distance vector routing have the strict objective to minimize the routing costs. This is done by assigning a cost to each link and minimizing the sum of the costs when calculating routes. The objective of field-based routing is quite different. Instead of calculating a path which strictly minimizes the routing cost, we also route packets in the direction of the highest service concentration. The motivation behind this design choice is to increase the probability of successful packet delivery as we have shown in the previous chapters.

6.5.4 Service Discrimination and Load Balancing

The choice of using field-based routing for accessing a service makes it easy to accommodate to the current load of close-by services. The capacity of service Q of each service in a STID field is used as an indicator for the load or capacity of the service. A service with high capacity/load uses a large value of Q, and thus generates an elevated potential distribution in its proximity. As a consequence, more packets are attracted towards this service instance. Balancing client load is done by varying the value of Q in accordance to the current load of a service. A service overloaded by client requests gradually reduces Q. As a result, the potential around the overloaded service decreases and packets are routed to alternative service instances. As soon as the load is tolerable, the original potential is reestablished by increasing Q.

6.5.5 Scalability

STID routing fields are maintained pro-actively by flooding packets from the services. Extensive flooding is always a major scalability concern. However, only STID routing-fields must be established pro-actively. Therefore, the number of pro-actively flooded messages is proportional to the number of services instances. We expect the number of service instances to be significantly smaller than the overall number of nodes in the network. UID routing fields are less critical since they only have to be maintained for the duration of active communication and are established in a limited scope around the client and the service.

To further improve scalability, we proposed to limit the distribution of routing field information of a service if its influence on the potential field drops below a certain threshold value. Assuming that the service instances are well distributed within the network, this mechanism has the effect of reducing the flooding of advertisements to a local scope.

6.5.6 Security

A major security threat of IP networks are Denial of Service (DoS) attacks. Our architecture mitigates this kind of threat. First, the routing state is limited in scope. UID routing state is established only in a restricted range between clients and services, and STID fields allow a sender to send packets only to one service (the one on the steepest gradient of the field). Thus, an attacker would have to place himself at strategic locations to be able to send unwanted traffic to a selected victim. This is a much higher burden as compared to IP networks, where a host can send packets to any other hosts in the network. Also, unwanted traffic addressed to a specific UID can be mitigated by changing the UID in use, since UIDs are created locally and on the fly. Thus, a victim host is able to create a new UID and communicates this one to the hosts with which it currently communicates.

6.6 Conclusions

This chapter proposes a new architecture for autonomic and situated communication in mobile environments. The architecture is service-centric by naming services and clients instead of network interfaces and considers mobility by design. The name space consists of location-independent identifiers which allows entities to keep their identity while moving. Also, we provide a set of identifiers (STIDs) which allows for clients communicating with services without binding themselves to specific service instances. Thus, the position of the client determines the optimal service. We also revise the routing mechanisms of IP routing to increase the robustness. By forwarding data packets in the direction of the highest service concentration, we increase the probability that packets reach a service when nodes are mobile.

There are still open research problems with regard to our architecture. We investigated communication between a client and

a service. Different communication patterns such as for example multicast have been left out. It would be interesting to see if it is possible to achieve more complex patterns on an anycast routing substrate.

Chapter 7

Conclusions and Future Work

In this chapter, we conclude our work on field-based routing. We first summarize our main results and contributions. Then, we mention possible weaknesses and shortcomings of our work. Following, we describe two new research projects that plan to use field-based routing in the context of sensor networks and discuss how they plan to extend the approach. Finally, we discuss still open research issues that we think deserve further attention.

7.1 Summary and Conclusions

In this thesis, we proposed field-based routing, a routing model inspired from the physical model of fields. We showed that the model is flexible and yet powerful by using it to solve different problems. We see the beauty of the model in its simplicity and clarity in design. The model describes routing in a very intuitive way. Furthermore, the model is easy to visualize and implement.

A major feature of field-based routing is its robustness in the face of link failures and node mobility. Field-based routing is able

to route data around broken links and/or to alternative entities with only local information. We also point out the limitations of the model. For k smaller than a lower bound we derive in Chapter 3, the routing might fail to converge due to local maxima in the potential fields. However, we show that these particular cases are more of academic interest and that in practical networks, the probability is very low that local maxima occur in the potential field.

Field-based routing incorporates a fundamental routing tradeoff between density and proximity. Consequentially, we analyze the impact of this tradeoff for anycast routing in wireless networks. We conclude that proximity-based anycast routing performs best in static networks. However, density-based anycast routing outperforms proximity-based routing in mobile and noisy environments.

Our field-based model is also able to differentiate entities based on their capacities. We exploit this additional property for building a distributed service discovery protocol for mobile ad hoc networks that is both distance- and capacity-aware. In a simplified way, this means that a close service instance is preferred over a distant one unless it has a higher capacity. At the same time, our protocol profits from the density-based routing properties of field-based routing to achieve a high degree of robustness.

Finally, we investigate how a network architecture for mobile ad hoc networks should be designed. Motivated by the limitations of the existing IP-based protocols, we propose a novel architecture based on field-based routing. The main challenge is to support one-to-one and one-to-any communication patterns on the top of the field-based routing substrate which is by design any-to-any (anycast). We define the required protocols to achieve those type of communication patterns and show (i) the benefits of our architecture versus IP and (ii) that our architecture is feasible by implementing a proof-of-concept prototype.

7.2 Weaknesses and Shortcomings

We have used field-based in the context of anycast routing and service discovery. In fact, these are the application scenarios were field-based routing has clear benefits. We have shown that field-based routing can also be used for unicast routing by defining groups with a single entity. However, we do not expect field-based routing to outperform other schemes in that context.

The communication paradigm we propose in this thesis has different semantics than IP, and is thus not compatible. For our approach to be deployed, this would required additional standardization and implementation efforts. Also, additional efforts will be needed to interconnect both types of technologies as we do not want to exclude clients in our system to access IP services and vice versa.

7.3 Future Work

7.3.1 The "Safety Critical Sensor Networks for Building Applications" Project

This research project is a KTI project including ETH Zurich, CSEM and Siemens Schweiz AG. The main goal of the project is to build a fire detection system based on wireless fire detection sensor nodes. The business idea of the project is to save costs of deployment by replacing the wiring with a wireless network.

This application is unique in its genre by the reliability requirements of the communication infrastructure: every false alarm leads to an unnecessary request to the fire department and a missed alarm might cost human life. The project plans to use field-based routing as robust routing mechanism. The aim of the project is to adapt field-based routing to the application specific requirements of such a system. The two main aspects are:

- How to run field-based routing on top of a MAC layer which is optimized for energy-efficiency of sensor nodes such as WiseMAC [**?**]?
- How to include additional metrics beyond the hop count such as for example the received signal strength indicator (RSSI) into the potential field.

7.3.2 The "RoadSens" Project

RoadSens is a joint research project between ETH Zurich and the Indian Institute of Science, Bangalore, India which is funded by the Indo-Swiss Joint Research Programme. The project deals with routing and addressing issues of sensor networks. The project investigates different aspects related to field-based routing:

- How to include additional metrics beyond the hop count such as for example the received signal strength indicator (RSSI), the energy level of individual nodes, or security metrics into the potential field.
- How to achieve data fusion when using a potential field?
- How to implement field-based routing on resource-limited devices? The main limitations are the limited energy budged of battery-powered sensor nodes and the limited computing capabilities.

7.3.3 Other Issues

In our opinion, in addition to the abovementioned projects, further research should be directed at the issues listed below:

- How could clients influence the potential fields to achieve for example load balancing. Clients could be modelled as negative point charges and advertise their presence similar

to how services do. Alternatively, the data packets could reduce the potential values of the nodes they traverse.

- It should be investigated if the model can be extended to consider the diversity of paths/links towards entities instead of considering the density of the entities. We believe that such an approach would also improve the robustness of the routing performance.

- We have shown that the probability of having a local maximum in the potential field is low, and thus did not put more efforts on solving this problem. To solve this problem, we could for example try to extend the link reversal algorithms proposed in [**?**] to convert a destination-disoriented DAG into a destination-oriented DAG. The research problem would be to assess if the proposed algorithm works when the DAG has many destinations (as it is the case with field-based routing), or to propose a new approach.

Appendix A

This appendix provides extensions for the proofs in Chapter 4, and sample values for the derived bound μ and ε in the same chapter.

A.1 Extension of Lemma 3

The condition for shortest path routing is:

$$\frac{1}{(\frac{D-1}{2}-1)^k}+\frac{N-1}{(\frac{D-1}{2}+2)^k}>\frac{1}{(\frac{D-1}{2}+1)^k}+\frac{N-1}{(\frac{D-1}{2})^k} \tag{A.1}$$

This inequality cannot be solved analytically for k. Before solving it numerically, we have to bring it in a cancellation-free form where the terms do not cancel each other. Therefore, we first separate k and N:

$$N-1<\frac{\frac{1}{(\frac{D-1}{2}+1)^k}-\frac{1}{(\frac{D-1}{2}-1)^k}}{\frac{1}{(\frac{D-1}{2}+2)^k}-\frac{1}{(\frac{D-1}{2})^k}}=f(k) \tag{A.2}$$

The function $f(k)$ can also be written as:

$$f(k)=\frac{(\frac{D-1}{2}+1)^{-k}-(\frac{D-1}{2}-1)^{-k}}{(\frac{D-1}{2}+2)^{-k}-(\frac{D-1}{2})^{-k}} \tag{A.3}$$

By using the relationship $a^{-k} - b^{-k} = 2(ab)^{-\frac{k}{2}} \sinh(\frac{k}{2} \log \frac{b}{a})$ that is derived with the definition of sinh, we transform $f(k)$ as:

$$f(k) = \frac{2 \cdot \left((\frac{D-1}{2} + 1)(\frac{D-1}{2} - 1) \right)^{-\frac{k}{2}} \cdot \sinh(\frac{k}{2} \log \frac{\frac{D-1}{2}-1}{\frac{D-1}{2}+1})}{2 \cdot \left((\frac{D-1}{2} + 2)(\frac{D-1}{2}) \right)^{-\frac{k}{2}} \cdot \sinh(\frac{k}{2} \log \frac{\frac{D-1}{2}}{\frac{D-1}{2}+2})} \tag{A.4}$$

or

$$f(k) = \left(\frac{(\frac{D-1}{2} + 2)\frac{D-1}{2}}{(\frac{D-1}{2} + 1)(\frac{D-1}{2} - 1)} \right)^{\frac{k}{2}} \cdot \frac{\sinh\left(\frac{k}{2} \log \frac{\frac{D-1}{2}-1}{\frac{D-1}{2}+1}\right)}{\sinh\left(\frac{k}{2} \log \frac{\frac{D-1}{2}}{\frac{D-1}{2}+2}\right)} \tag{A.5}$$

In this form, it is now possible to find numerically the solution for $f(k) > N - 1$ (e.g., using the secant method).

A.2 Extension of Lemma 4

The condition that the steepest gradient does not point to next hop on the shortest path is:

$$1 + \frac{N-1}{(D-1)^k} \leq \frac{1}{3^k} + \frac{N-1}{(D-3)^k} \tag{A.6}$$

Again, this inequality cannot be solved analytically for k, and solving it numerically is problematic since the form is not cancellation-free. Therefore, we bring it to a form where the terms do not cancel each other. We first separate k and N:

$$N - 1 \geq \frac{1 - \frac{1}{3^k}}{\frac{1}{(D-3)^k} - \frac{1}{(D-1)^k}} = f(k) \tag{A.7}$$

The function $f(k)$ can be written as:

$$f(k) = \frac{1^{-k} - 3^{-k}}{(D-3)^{-k} - (D-1)^{-k}} \tag{A.8}$$

By using the relationship $a^{-k} - b^{-k} = 2(ab)^{-\frac{k}{2}} \sinh(\frac{k}{2} \log \frac{b}{a})$, we get

$$f(k) = \frac{2 \cdot 3^{-\frac{k}{2}} \sinh(\frac{k}{2} \log 3)}{2 \cdot \big((D-1)(D-3)\big)^{-\frac{k}{2}} \sinh(\frac{k}{2} \log \frac{D-1}{D-3})} \tag{A.9}$$

or

$$f(k) = \Big(\frac{(D-1)(D-3)}{3}\Big)^{\frac{k}{2}} \frac{\sinh(\frac{k}{2} \log 3)}{\sinh(\frac{k}{2} \log \frac{D-1}{D-3})} \tag{A.10}$$

It can be shown that for $D > \frac{1+\sqrt{13}}{2} \approx 2.3$, $f(k)$ is monotonously increasing for any $k > 0$. With this restriction for D, which means that the network diameter must at least be 3, $N - 1 \geq f(k)$ has valid solutions if $N - 1 > \frac{\log 3}{\log \frac{D}{D-1}}$. The solutions can again be found with e.g. the secant method.

D / N	6	8	10	12	14	16	18	20
4	1.795635	2.931176	4.049784	5.161040	6.268394	7.373423	8.476947	9.579444
6	2.936760	4.591675	6.224987	7.849220	9.468746	11.085512	12.700515	14.314324
8	3.657162	5.652006	7.623068	9.584353	11.540646	13.494042	15.445605	17.395938
10	4.183686	6.431778	8.654766	10.867593	13.075283	15.280017	17.482893	19.684523
12	4.598563	7.048652	9.472706	11.886338	14.294741	16.700152	19.103687	21.505970
14	4.940854	7.559027	10.150448	12.731245	15.306746	17.879221	20.449810	23.019141
16	5.232170	7.994312	10.729110	13.453120	16.171770	18.887369	21.601069	24.313505
18	5.485726	8.373796	11.234018	14.083305	16.927177	19.767971	22.606854	25.444466
20	5.710183	8.710176	11.681874	14.642505	17.597668	20.549730	23.499864	26.448721
22	5.911535	9.012257	12.084283	15.145120	18.200436	21.252628	24.302877	27.351841
24	6.094092	9.286393	12.449630	15.601562	18.747926	21.891138	25.032397	28.172357

Table A.1: *Sample values for μ. N is the number of anycast group members and D is the network diameter.*

D / N	6	8	10	12	14	16	18	20
6	-	-	-	-	-	-	-	-
8	0.129472	-	-	-	-	-	-	-
10	0.343468	0.061210	-	-	-	-	-	-
12	0.511681	0.197150	0.031419	-	-	-	-	-
14	0.650004	0.309386	0.129192	0.015418	-	-	-	-
16	0.767310	0.404871	0.212506	0.090838	0.005733	-	-	-
18	0.869051	0.487902	0.285049	0.156559	0.066601	-	-	-
20	0.958808	0.561315	0.349261	0.214772	0.120540	0.050101	-	
22	1.039066	0.627080	0.406841	0.267005	0.168956	0.095628	0.038259	-
24	1.111609	0.686620	0.459017	0.314358	0.212865	0.136930	0.077500	0.029415

Table A.2: *Sample values for* ε*. N is the number of anycast group members and D is the network diameter.*

Bibliography

[80297] IEEE Std 802.11-1997. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. Number ISBN 1-55937-935-9. 1997.

[ABB+04] Daniel Aguayo, John Bricket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of ACM SIGCOMM'04*, Portland, Oregon, USA, October 2004.

[And03] D. G. Anderson. Mayday: Distributed Filtering for Internet Services. In *USENIX Symposium on Internet Technologies and Systems*, Seattle, WA, USA, March 2003.

[AWSL99] William Adjie-Winoto, Elliot Schwartz, and Jeremy Lilley. The Design and Implementation of an Intentional Naming System. In *Proceedings of the 17th Symposium on Operating Systems Principles (SOSP '99)*, pages 186–201, Charleston, SC, USA, 1999.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475, December 1998.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF RFC 1633, June 1994.

[BF05] Hitesh Ballani and Paul Francis. Towards a Global IP Anycast Service. In *Proc. of ACM SIGCOMM*, Philadelphia, USA, August 2005.

[BLR03] Anindya Basu, Alvin Lin, and Sharad Ramanathan. Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm. In *Proceedings of the ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM'03)*, Karlsruhe, Germany, August 2003.

[BLR+04] Hari Balakrishnan, Karthik Laksminarayanan, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Michael Walfish. A Layered Naming Architecture for the Internet. In *Proceedings of the ACM SIGCOMM*, pages 343–353, Portland, Oregon, USA, August 2004.

[BRS03] Christian Bettstetter, Giovanni Resta, and Paolo Santi. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.

[CBFP03] David Clark, Robert Braden, Aaron Falk, and Venkata Pingali. FARA: Reorganizing the Addressing Architecture. In *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 313–321, Karlsruhe, Germany, August 2003.

[CBR05] I. Chakeres and E. Belding-Royer. Dynamic MANET On-demand (DYMO) Routing. draft-ietf-manet-dymo-03.txt, October 2005.

[CCS96] I. Castineyra, N. Chiappa, and M. Steenstrup. The Nimrod Routing Architecture. IETF RFC 1992, August 1996.

[CD97] G. Di Caro and M. Dorigo. AntNet: A Mobile Agents Approach to Adaptive Routing. Technical Report IRIDIA/97-12, Université Libre de Bruxelles, Belgium, 1997.

[CDE+05] David Culler, Prabal Dutta, Cheng Tien Ee, Rodrigo Fonseca, Jonathan Hui, Philip Levis, Joseph Polastre, Scott Shenker, Ion Stoica, Gilman Tolle, and Jerry Zhao. Towards a Sensor Network Architecture: Lowering the Waistline. In *Proceedings of the International Workshop on Hot Topics in Operating Systems (HosOS)*, Santa Fe, NM, USA, June 2005.

[CEH+01] A. Cerpa, J.E. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod. Habitat Monitoring: Application Driver for Wireless Communications Technology. *ACM SIGCOMM Computer Communication Review*, 31(2):20–41, April 2001.

[CGT05] Marco Conti, E. Gregori, and G. Turi. A Cross Layer Optimization of Gnutella for Mobile Ad Hoc Networks. In *Proccedings of ACM MobiHoc Symposium)*, Urbana-Champain, May 2005.

[CJ03] Thomas Clausen and Philippe Jacquet. Optimized Link State Routing Protocol. IETF Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003.

[CK74] V. Cerf and R. Kahn. A Protocol for Packet Network Interconnection. *IEEE Transactions on Communication*, 22:637–648, May 1974.

[CZH+99] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R. Katz. An Architecture for a Secure Service Discovery Service. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, pages 24–35, Seattle, Washington, USA, August 1999.

[DÖ5] Thomas Dübendorfer. *Impact Analysis, Early Detection and Mitigation of Large-Scale Internet Attacks*. PhD thesis, ETH, Zurich, Switzerland, December 2005.

[Del05] Franca Delmastro. From Pastry to CrossROAD: CROSS-Layer Ring Overlay for AD Hoc Networks. In *PerCom Workshops*, Hawaii, March 2005.

[DH98] S. Deering and R. Hinden. IP Version 6 Addressing Architecture. IETF RFC 2373, July 1998.

[EHD04] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of InfrastructureWireless Sensor Networks. In *Proceedings og the Ninth IEEE Symposium on Computers and Communication (ISCC)*, pages 244–251, Alexandria, Egypt, June 2004.

[FG01] Paul Francis and R. Gummadi. IPNL: A NAT-extended Internet Architecture. In *ACM SIGCOMM*, San Diego, CA, USA, August 2001.

[fre] freifunk.net. http://www.freifunk.net.

[GB81] E. Gafni and D. Bertsekas. Distributed Algorithms for Generating Loop-free Routes in Networks with Frequently Changing Topology. *IEEE Transactions on Communications*, 29(1):11–15, 1981.

[GC00] M. Gritter and D. R. Cheriton. A new Next Next-generation Internet Architecture. http://www-dsg.stanford.edu/triad/, July 2000.

[Gro99] The Bluetooth Special Interest Group. Specification of the Bluetooth system: Part B Baseband Specification, December 1999.

[HDVL03] Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark - A Service Discovery and Delivery Protocol for Ad-Hoc Networks. In *Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, USA, March 2003.

[HPS02] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samae. Zone Routing Protocol (ZRP). IETF Internet Draft, draft-ietf-manet-zrp-04.txt, July 2002.

[IGE00] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the ACM/IEEE Internation Conference on Mobile Computing and Networking (Mobicom '00)*, Boston, USA, 2000.

[IL99] Chalermek Intanagonwiwat and Dante De Lucia. The Sink-Based Anycast Routing Protocol for Ad Hoc Wireless Sensor Networks. Technical Report 99-698, USC Computer Science, CA, USA, 1999.

[Inc00] TIBCO Software Inc. TIB/Rendezvous Concepts. Technical Report Release 6.4, Palo Alto, CA, October 2000.

[JMHJ02] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva. The Dynamic Source Routing

Protocol for Mobile Ad Hoc Networks (DSR). February 2002.

[JOW+02] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, CA, USA, October 2002.

[KMR02] A.D. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002.

[KP02] Rajeev Koodli and Charles Perkins. Service Discovery in On-Demand Ad Hoc Networks. IETF Internet Draft draft-koodli-manet-servicediscovery-00.txt, October 2002.

[KT03] Ulas C. Kozat and Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of the IEEE INFOCOM*, San Francisco, USA, April 2003.

[KV98] Y.-B. Ko and N. H. Vaidya. Location-aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proceedings of the Fourth ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pages 66–75, Dallas, Texas, USA, 1998.

[KW00] Dina Katabi and John Wroclawski. A Framework for Scalable Global IP-Anycast (GIA). In *Proc. of ACM SIGCOMM*, Stockholm, Sweden, August 2000.

[LMP03] T. Lin, S. F. Midkiff, and J. S. Park. Mobility versus Link Stability in Simulation of Mobile Ad Hoc Networks. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 3–8, 2003.

[LMP05a] Vincent Lenders, Martin May, and Bernhard Plattner. Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Taormina, Italy, June 2005.

[LMP05b] Vincent Lenders, Martin May, and Bernhard Plattner. Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach. *Elsevier Journal on Pervasive and Mobile Computing*, 1(3), September 2005.

[LMP05c] Vincent Lenders, Martin May, and Bernhard Plattner. Service Discovery on Dynamic Networks. EU PCT Patent, EP No. 04030863.7, February 2005.

[LMP05d] Vincent Lenders, Martin May, and Bernhard Plattner. Towards a New Communication Paradigm for Mobile Ad Hoc Networks. In *Proceedings of the IEEE nternational Workshop on Heterogeneous Multi-Hop Wireless and Mobile Networks (MHWMN'05)*, Washington DC, USA, November 2005.

[LMP06] Vincent Lenders, Martin May, and Bernhard Plattner. Density-based vs. Proximity-based Anycast Routing for Mobile Networks. In *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[Mal98] G. Malkin. Rip version 2. IETF RFC 2453, November 1998.

[man04] IETF Mobile Ad-hoc Networking (manet) Working Group. http://www.ietf.org/html.charters/manet-charter.html, 2004.

[MC04] Joseph P. Macker and M. Scott Corson. *Mobile Ad Hoc Networks (MANETs): Routing Technology for Dynamic Wireless Networking*, chapter 9, pages 255–274. IEEE Press, 2004.

[Mic03a] Microsoft. The Universal Plug and Play (UPnP) Forum. http://www.upnp.org, 2003.

[Mic03b] Sun Microsystems. Jini Architecture Specification. version 2.0, June 2003.

[MNH04] R. Moslowitz, P. Nikander, and T. Henderson. Host Identity Protocol. draft-ietf-hip-base-01, IETF Internet draft, October 2004. work in progress.

[Moy98] J. Moy. OSPF Version 2. IETF RFC 2328, April 1998.

[NBBB98] K. Nichols, S. Blake, F Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474, December 1998.

[NC04] W. Navid and T. Camp. Stationary Distributions for the Random Waypoint Model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[nyc] Nycwireless. http://www.nycwireless.com.

[OLT03] R. Ogier, M. Lewis, and F. Templin. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). IETF Internet Draft, draft-ietf-manet-tbrpf-09.txt, June 2003.

[PB94] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Comp. Commun. Rev.*, pages pp 234–44, October 1994.

[PBRD02] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Internet Draft, draft-ietf-manet-aodv-12.txt, November 2002.

[PC97] V. Park and M.S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[Per96] Charles Perkins. IP Mobility Support. IETF RFC 2002, October 1996.

[PHL+05] Joseph Polastre, Jonathan Hui, Philip Levis, Jerry Zhao, David Culler, Scott Shenker, and Ion Stoica. A Unifying Link Abstraction for Wireless Sensor Networks. In *ACM SenSys*, San Diego, CA, USA, November 2005.

[PM99a] V. Park and J. Macker. Anycast Routing for Mobile Services. In *Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, March 1999.

[PM99b] Vincent D. Park and Joseph P. Macker. Anycast Routing for Mobile Networking. In *Proc. of MILCOM*, Atlantic City, New Jersey, USA, November 1999.

[PMM93a] C. Partidge, T. Mendez, and W. Milliken. Host Anycasting Service. IETF RFC 1546, November 1993.

[PMM93b] C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service. IETF RFC 1546, November 1993.

[QVL01] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2001).*, 2001.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protcol 4 (BGP-4). IETF RFC 1771, March 1995.

[RSM05] Sylvia Ratnasamy, Scott Shenker, and Steven McCanne. Towards an Evolvable Internet Architecture. In *SIGCOMM*, Philadelphia, USA, August 2005.

[SAZ+02] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet Indirection Infrastructure. In *Proceedings of ACM SIGCOMM Conference*, Pittsburgh, PA, USA, August 2002.

[SB00] A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *ACM MOBICOM*, Boston, Massachusetts, USA, August 2000.

[sea] Seattle wireless. http://www.seattlewireless.net.

[TT05] S. Toumpis and L. Tassiulas. Packetostatics: Deployement of Massively Dense Sensor Networks as an Electrostatic Problem. In *IEEE INFOCOM*, Miami, USA, March 2005.

[VGPK97] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service Location Protocol. RFC 2165 (http://www.ietf.org/rfc/rfc2165.txt), June 1997.

[Wan03] Xiaowei Wang. NIRA: A new Internet Routing Architecture. In *ACM SIGCOMM Workshop on Future Directions in Network Architectures*, Karlsruhe, Germany, August 2003.

[WZJ03a] Jianxin Wang, Yuan Zheng, and Weijia Jia. A-DSR: A DSR-based Anycast Protocol for IPv6 Flow In Mobile Ad Hoc Networks. In *Proc. of the IEEE Vehicular Technology Conference*, Orlando, Florida, USA, October 2003.

[WZJ03b] Jianxin Wang, Yuan Zheng, and Weijia Jia. An AODV-based Anycast Protocol in Mobile Ad Hoc Network. In *Proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, Beijing, China, September 2003.

[ZAFB00] Ellen W. Zegura, Mostafa H. Ammar, Zongming Fei, and Samrat Bhattacharjee. Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service. *IEEE/ACM Transactions on Networking*, 8(4), 2000.

[ZBG98] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS '98)*, Banff, Alberta, Canada, May 1998.

Acknowledgements

A number of people contributed to this thesis in their own unique ways and I would like to express my gratitude to them.

First, I would like to thank my advisor Prof. Dr. Bernhard Plattner for his continuous support and the inspiring discussions throughout my time in his lab. I really enjoyed the working atmosphere and the possibility to explore and work on a variety of exciting research topics.

I am also very grateful to my co-advisors Prof. Dr. Jim Kurose from the University of Massachusetts at Amherst and Prof. Marco Conti from CNR, Italy for very thoroughly reviewing this thesis. Their precise comments and suggestions greatly improved this dissertation.

I am deeply grateful to Dr. Martin May for his dedicated commitment and overwhelming support. Among the infinite number of things I learned from him was how to write papers, and I will never forget the hours we spent together while talking over a draft of a paper and getting the most valuable feedback I can think of.

Thanks to Dr. Ulrich Fiedler and Dr. Christan Bruns for their precious comments and feedback on the modelling aspects of field-based routing.

Thanks also to Prof. Dr. Polly Huang who convinced me after my graduation to do a PhD and initiated me to research when

she was working in our lab.

Thanks to Prof. Dr. Gunnar Karlsson from KTH Stockholm with whom I had the opportunity to work with during his sabbatical year in our group. His most interesting and original perspectives widened my mindset.

Thanks to Simon Heimlicher and Georgios Parissidis for choosing me as their PhD mentor and for the excellent mentoring dinners. I hope they enjoyed the collaboration as much as I did.

I also want to thank the my colleagues from the lab Rainer Baumann, Marcel Baur, Matthias Bossardt, Daniela Brauckhoff, Thomas Dübendorfer, Karoly Farkas, Stefan Frei, Jan Gerke, Hasan Hasan, David Hausheer, Kostantinos Katrinis, Ralph Keller (my office mate during the first two years), Jan Mischke, Marc Rennhard, Catrina Sposato, Mario Strasser, Bernhard Tellenbach, Arno Wagner, and Nathalie Weiler for the great time we had together.

I would also like to acknowledge the contributions from students doing a semester or master thesis under my supervision: Christian Braun, Katrin Bretscher, Lorenz Bührer, Axel Burri, Nicolas Cedraschi, Bernhard Distl, Daniel Grob, Nicole Hatt, Christian Hess, Samuel Kasper, Oliver Keiser, Philipp Minnig, Sandro Schifferle, Diana Senn, Philipp Sommer, Florian Süss, Ivo Trajkovic, Clemens Wacha, Jörg Wagner, Andreas Weibel, Mario Widmer, and Lukas Winterhalder.

Finally, I would like to thank my wife Nadja and my parents for their unconditional support over the whole years as well as my dog Feno for the relaxing and inspiring walking opportunities.

Curriculum Vitae

Vincent Lenders was born in Berchem-Ste-Agathe, Belgium, on October 19, 1977. He attended primary school in La Hulpe, Belgium. In 1988, he moved to Basel and began his high school education. After an exchange year in 1995 in New Jersey, USA, he obtained his Matura degree in mathematics and natural sciences (type C) in 1996.

His general interest in technical sciences motivated him to study electrical engineering at ETH Zurich. During his studies, Vincent Lenders worked part-time as a system administrator at Arthur. D. Little in Thalwil, as a teaching assistant at ETH Zurich, and as an intern at ABB, Mannheim. In spring 2001, he obtained his Masters degree (Dipl. El.-Ing. ETH).

Directly after his graduation, he joined the Computer Engineering and Networks Laboratory at ETH Zurich as a PhD student. There, he worked on various research projects in the field of mobile, wireless, and ad hoc networking, and was also active in various teaching activities.

After finishing his PhD in 2006, he joined Princeton University, USA as a post-doctoral research fellow.