HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Electrical and Communications Engineering

Olli Apilo

Performance of Randomized Forwarding Methods in Large Ad Hoc Networks

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Supervisor: Professor Jorma Virtamo
Instructor: D.Sc.(Tech.) Pasi Lassila

Espoo, 1st February 2006

Author: Olli Apilo

Title: Performance of Randomized Forwarding Methods in Large Ad Hoc Networks

Date: 1st February 2006 **Number of pages:** 103

Department: Department of Electrical and Communications Engineering

Professorship: S-38 Teletraffic Theory

Supervisor: Professor Jorma Virtamo

Instructor: D.Sc.(Tech.) Pasi Lassila

An ad hoc network is a wireless network consisting of mobile nodes that communicate with each other in a multihop fashion without any need for fixed infrastructure or centralized control. Most of the factors that affect the performance of an ad hoc network are related to routing and medium access control (MAC). In this study, different routing and MAC methods for ad hoc networks are surveyed and a more detailed study is made on the performance of different forwarding methods in a large ad hoc network.

In this latter part of the study, we model a large ad hoc network. In this setting, the routing problem can be decoupled into a macroscopic and a microscopic level. At the microscopic level, the direction of packet flow is given by the macroscopic level routing algorithm. Thus, the task of the microscopic level forwarding method is to maximize the flow of packets in the given direction. The packet flow intensity depends only on the slotted ALOHA parameter for the transmission probability and the network density. We study the maximum packet flow intensity by simulations with respect to these two parameters and compare the performance of four different forwarding methods, namely one deterministic and three randomized forwarding methods.

The randomized forwarding methods achieved better performance than deterministic forwarding. Deterministic forwarding concentrated traffic on a few paths leaving most of the network under-utilized, while randomized forwarding spread traffic to the network more efficiently. Of the randomized methods, opportunistic forwarding, which modifies the MAC protocol to co-operate with forwarding rules, was clearly the best.

Keywords: Ad hoc networks, routing, medium access control, performance evaluation, density of progress

Tekijä: Olli Apilo

Otsikko: Satunnaistettujen välitysmenetelmien suorituskyky suurissa ad hoc -verkoissa

Päivämäärä: 1. helmikuuta 2006 **Sivumäärä:** 103

Osasto: Sähkö- ja tietoliikennetekniikan osasto

Professuuri: S-38 Teleliikenneteoria

Työn valvoja: Professori Jorma Virtamo

Työn ohjaaja: TkT Pasi Lassila

Ad hoc -verkko on langaton verkko, joka koostuu liikuteltavista päätelaitteista. Päätelaitteet voivat viestiä keskenään monihyppyisesti ilman kiinteää verkkoinfrastruktuuria ja keskitettyä valvontaa. Useimmat ad hoc -verkon suorituskykyyn vaikuttavista tekijöistä liittyvät reititykseen ja pääsynvalvontaan (MAC). Tässä työssä tehdään kirjallisuustutkimus ad hoc -verkkojen reititys- ja MAC-menetelmistä. Lisäksi tutkitaan yksityiskohtaisemmin eri välitysmenetelmien suorituskykyä suuressa ad hoc -verkossa.

Tässä työn jälkimmäisessä osassa mallinnetaan suuri ad hoc -verkko. Suuressa ad hoc -verkossa reititysongelma voidaan jakaa makroskooppiselle ja mikroskooppiselle tasolle. Mikroskooppisella tasolla pakettivuon suunta saadaan annettuna makroskooppisen tason reititysalgoritmilta. Mikroskooppisen tason välitysmenetelmän tehtävänä on maksimoida pakettivuo annettuun suuntaan. Pakettivuon tiheys riippuu vain Slotted ALOHAn lähetystodennäköisyysparametrista ja verkon tiheydestä. Maksimaalista pakettivuon tiheyttä tutkitaan näiden kahden parametrin suhteen ja vertaillaan neljän eri välitysmenetelmän, yhden deterministisen ja kolmen satunnaistetun, suorituskykyä.

Satunnaistetut välitysmenetelmät saavuttivat paremman suorituskyvyn kuin deterministinen välitys. Deterministinen välitys keskitti liikenteen muutamille poluille jättäen suurimman osan verkosta hyödyntämättä, kun taas satunnaistettu välitys levitti liikenteen verkkoon tehokkaammin. Satunnaistetuista menetelmistä opportunistinen välitys, joka yhdistää MAC-protokollan ja välityksen toiminnan, oli selvästi paras.

Avainsanat: Ad hoc verkot, reititys, MAC, suorituskyvyn arviointi, etenemisen tiheys

Acknowledgements

This Master's thesis was written in the Networking Laboratory of Helsinki University of Technology for the NAPS project funded by the Academy of Finland.

I would like to thank the supervisor of the thesis Professor Jorma Virtamo for his invaluable guidance and the instructor of the thesis D.Sc.(Tech.) Pasi Lassila for all the help and patience during the whole thesis process. Special thanks go to D.Sc.(Tech.) Esa Hyytiä for valuable instructions during the early work and to Olli-Pekka Lamminen for helping me with debugging problems.

I would also like to thank all the people at the Networking Laboratory for providing such an inspirational working atmosphere.

I am grateful to my parents for their support during my studies and to my friends for all the relaxing and hilarious moments. Finally, I would like to thank Pälvi for her loving support and for motivating me in my studies.

Contents

Chapter 1

Introduction

1.1 Background

Ad hoc networking is one of the most active fields in communications research today. The idea of ad hoc networking is not actually new; the research has been going on for over 30 years. The first ad hoc network applications were designed for military environment but during the last 10 years the interest in commercial ad hoc networks has been growing due to the emergence of inexpensive, lightweight wireless devices.

An ad hoc network consists of wireless nodes each having a certain communication range. An ad hoc node is able to communicate directly with another ad hoc node if it is within its communication range. The distinctive feature of ad hoc networks is that the communication between two nodes can still be possible in a multihop fashion if the direct communication is not possible. In multihop communications intermediate nodes act as routers forwarding data towards the destination. The nodes in ad hoc network can communicate without any fixed infrastructure and without centralized control (see Figure 1.1). An ad hoc network should adapt to changes in the network topology due to, e.g., node mobility, radio link failures and power control methods without any need for system administration.

The major advantages of ad hoc networking are its rapid deployment and its robustness and flexibility against changes in topology. These advantages are especially important in military and rescue applications such as battlefield and disaster area communications. The fact that there is no need for fixed infrastructure is also beneficial when setting up wireless communications in conferences, exhibitions and campus areas.

The performance of a network is usually given in terms of throughput, the

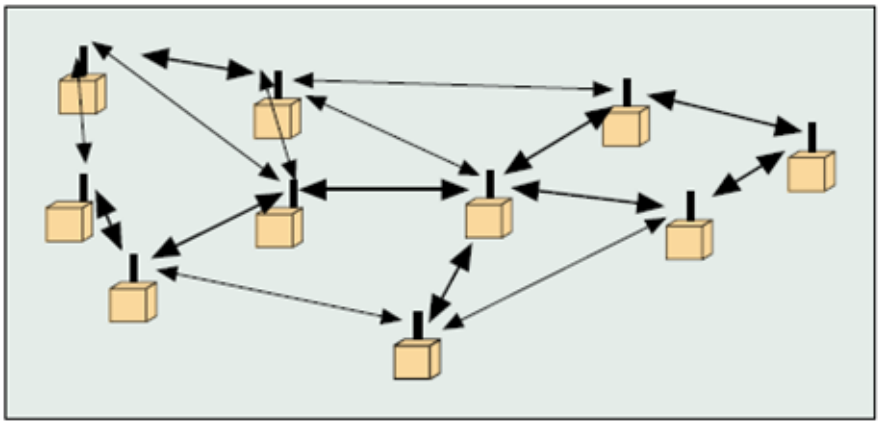


Figure 1.1: A collection of ad hoc nodes forming a network [1].

amount of data that can be transmitted over the network in a given time, and latency, the amount of time it takes for a message to travel from the source to the destination. In ad hoc networks, there are also other performance metrics, such as energy consumption in a given time and fairness among the nodes as they contend for the access to the network. Most factors that affect the performance of an ad hoc network are related to routing and medium access control (MAC). Often routing and MAC are interleaved so that in order to achieve the best performance for a given network and for a given application, they have to be jointly designed. This is commonly referred to as cross-layer design.

1.2 Objective of the study

The objective of this study is twofold. At first, we present a literature survey of current MAC and routing methods in ad hoc networks. The second part of the thesis is a simulation study that aims to maximize the network-wide throughput and compares the performance of geographic forwarding methods in a large ad hoc network.

The ad hoc networking research in the field of media access control and, especially, routing has been extremely active in the past 10 years. The number of related publications per year is ever increasing. Thus, we do not claim to have a comprehensive survey of all MAC and routing protocols. Instead, our approach is to establish a clear classification of these protocols and present only a few most important protocols or algorithms for each of these classes.

In the simulation part, we model an infinite, large ad hoc network and compare

the performance of different geographic forwarding methods combined with a simple slotted ALOHA MAC scheme. In large ad hoc networks, the routing problem decouples into two differently scaled problems; at the macroscopic level, the problem is to find route shapes that minimize a given cost metric and at the microscopic level, traffic flow to a given direction need to be maximized. We focus on the microscopic level aiming to find the optimal network density and the optimal slotted ALOHA transmission probability maximizing the traffic flow intensity in a given direction for different local forwarding rules.

1.3 Structure of the thesis

Chapter 2 is an introductory overview of ad hoc networks. We describe the history and challenges of ad hoc networking and present the common graph generalization for ad hoc networks. Finally, we introduce wireless sensor networks that are special cases of ad hoc networks.

In chapter 3, we make a survey of MAC protocols used in ad hoc networks. Before considering the problem of medium access, we present a few most common models for radio propagation and interference used in ad hoc networking research. For the rest of the chapter, we divide MAC protocols into traditional protocols used in early packet radio networks, general reservation-based protocols, power aware protocols, protocols using directional antennas, multiple channel protocols and protocols used in wireless sensor networks.

Chapter 4 surveys unicast routing protocols for ad hoc networks. At first, we introduce some optimizations to network-wide broadcasting or flooding that is an essential part of many unicast routing protocols. In the following sections, routing protocols are classified as proactive, reactive and geographic protocols. Finally, we give an overview of methods and issues specifically related to routing in wireless sensor networks.

Chapter 5 presents the general framework for the simulation study. A model for an infinite ad hoc network and a performance optimization criterion for it is presented. Then we describe the geographic routing algorithms used in simulations. Finally, related work on routing and MAC layer performance optimization in ad hoc networks is discussed.

Chapter 6 collects together the most relevant implementation issues and the results of the simulation. Chapter 7 concludes the study and proposals for further work are presented.

Chapter 2

Ad hoc networks

This chapter presents a brief overview of the history and the challenges of wireless ad hoc networks. In addition, the common graph abstraction for ad hoc networks is presented. Finally, an overview of wireless sensor networks is given as an example of a specialized ad hoc network.

2.1 History of ad hoc networking

The research on ad hoc networks was initiated in the early 1970s by Defense Advanced Research Projects Agency (DARPA). The initial testing of the new Packet Radio Network (PRNET) began in 1975 and the system overview was presented in 1977 [2]. The PRNET introduced multihop communications, a distributed distance-vector-based routing, a new medium access scheme Carrier Sense Multiple Access (CSMA) and allowed limited mobility for the nodes. All these improvements made PRNET superior to its predecessor, the ALOHA network [3].

During the 1970s and 1980s packet radio networks were mainly used for military purposes. The interest on commercial ad hoc networks began to grow in the early 1990s when notebook computers became popular. The term ad hoc network was used to describe a collection of relatively small mobile hosts, such as notebooks, communicating without any kind of fixed infrastructure. One of the earliest uses of this term was in [4] where Destination-Sequenced Distance-Vector (DSDV) routing protocol for ad hoc networks was presented. A little later ad hoc network became an "official" term when the IEEE 802.11 subcommittee on wireless local area networks adopted the term. Active research on ad hoc networks started and soon there was a need for ad hoc networking standardization. In 1997 the IETF Mobile Ad Hoc Networking (MANET) working

group was formed to specify standard interfaces and protocols for support of IP-based communications over ad hoc networks.

2.2 Challenges in ad hoc networking

Despite the long history of military packet radio networks, the research on commercial ad hoc networks is fairly young. The unique characteristics of ad hoc networks cause design challenges that differ from those of wired networks and cellular wireless networks. There are still numerous unsolved challenges in ad hoc networking. The most important challenges are related to the following:

- media access control,
- routing,
- energy efficiency and power control,
- quality of service, and
- security

In this section we present a brief overview of each of these challenges. There are a number of more thorough overviews available in survey papers. For example, see [5, 6] on media access control, [7, 8] on routing, [9] on power control, [10, 11] on quality of service and [12] on security.

The techniques that control access to the wireless medium are divided into two main groups: controlled access and random access. In controlled MAC protocols the medium is divided either statically or dynamically into channels and the access to the channels is usually centrally controlled. Cellular wireless networks typically use controlled MAC techniques such as TDMA, FDMA and CDMA. Because there is no centralized control and usually no global synchronization in ad hoc networks, random access MAC protocols are more suitable for ad hoc networking. In random access MAC protocols multiple nodes share the same frequency channel and the access to it is somehow random. These MAC protocols aim to guarantee a fair access to all nodes and to avoid collisions with other transmissions.

If measured in the number of journal and conference publications, routing is the most active research field in ad hoc networking. This is not surprising since node mobility as well as energy conservation and scalability requirements make efficient routing challenging. The distance-vector and link-state-based

routing protocols used in wired networks are usually too slow to react to frequent changes in ad hoc network topology. A good routing strategy efficiently uses the limited resources of an ad hoc network while at the same time dynamically adapts to changes in network conditions. In addition to these basic requirements, the routing protocol may have to fulfill e.g. a certain quality of service or security level requirements as well.

Ad hoc nodes are equipped with batteries that are responsible for providing power for communications. In addition to their own communication, ad hoc nodes consume their batteries while relaying data sent by other nodes. Thus, the battery lifetime sets constraints on the design of ad hoc networks. The simplest way to save the battery capacity of a node is to turn its power off during idle times (power management). Another simple approach is to vary the transmission power depending on the distance to the next hop receiver (power control) and prefer long chains of short hops to the final destination instead of fewer longer hops. However, the power control by varying the transmission power is a complex issue because the transmission power level affects the quality of the signal at the receiver, the range of transmission and the amount of interference caused to other receivers.

The network offering a certain Quality of Service (QoS) is expected to guarantee a certain performance level to users. The end-to-end performance is measured by service attributes such as delay, probability of packet loss, out-of-order delivery of packets and delay variation. In ad hoc networks, there can be additional QoS attributes like power consumption in a time period or the coverage of a given service. The mobile ad hoc network is a difficult environment for supporting QoS. The shared wireless channel is a very unpredictable medium because of random collisions and changes in radio link properties. QoS-aware routing would require the reservation of resources along the route, which is often impossible, if the node mobility is high.

Because ad hoc nodes use a shared frequency channel, each transmission can be considered as a broadcast to all nodes within the transmission range of the sender. This is potentially a very insecure environment for communications. The potential security threats include eavesdropping, impersonation and denial of service (DoS) attacks.

Of the above issues, MAC and routing protocols are covered in depth in Chapters 3 and 4, respectively. Issues related to power control, QoS and security are not considered in more detail since they are out of the scope of the present study.

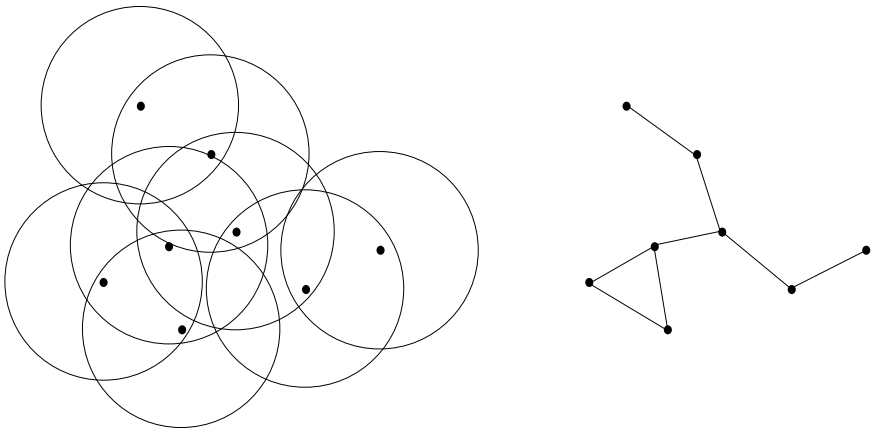


Figure 2.1: A group of ad hoc nodes with communication ranges shown and the corresponding network graph

2.3 Ad hoc network as a graph

Although the characteristics of a wireless medium vary dynamically, it is often useful to abstract away the physical layer details of an ad hoc network. This kind of abstractions are especially beneficial in the early phases of routing or MAC design and when the performance of different protocols are compared. The most common abstraction of ad hoc networking is to present the network as a graph.

An ad hoc network can be modeled as a directed graph. Each ad hoc node is considered as a vertex and there exists an edge AB from Node A to Node B if Node B is within the communication range of Node A. See figure 2.1 for an example network graph. The edges are not necessarily bidirectional as the radio link properties and the transmission powers can be different between the pair of nodes. A network graph is connected if there exists a path from every node to every other node. Correspondingly a network graph is k-connected if there exist k node disjoint paths between every pair of nodes. On the other hand, even if any $k-1$ nodes are removed from a k-connected network, the network still stays (1-)connected. The level of connectivity is related to the average node degree. A node degree gives the number of neighbors for the node and correspondingly in a general graph, a vertex degree is the number of edges with touch of the vertex.

The connectivity properties of ad hoc networks have a significant impact on their performance. The coverage of an ad hoc network is directly related to the number of connected nodes and the level of connectivity describes how reliable the network is. For analytical results related to the connectivity of ad hoc networks, see [13].

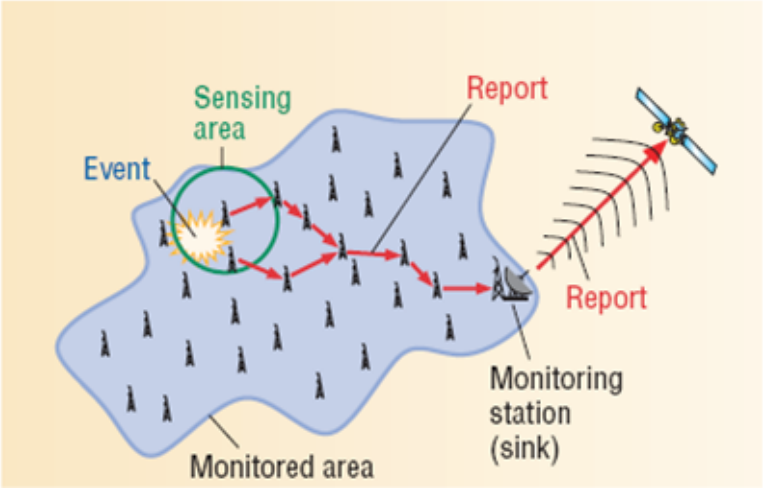


Figure 2.2: A sensor network formed to monitor an area [16].

2.4 Wireless sensor networks

In this section we give a brief overview of a specialized ad hoc network, a wireless sensor network. In addition to this overview, MAC and routing issues and protocols are later discussed in depth. The research on sensor networks is increasing and there are still many issues to be solved. For an overview of the whole research field, see for example [14, 15].

A wireless sensor network is a specialized ad hoc network consisting of a large number of small sensor nodes. The sensor nodes are miniature devices capable of sensing their environment, communicating with other sensor nodes and processing the sensed data. A sensor network can be set up by placing a set of sensor nodes randomly near the investigated phenomenon and the nodes collaboratively self-organize themselves into a sensor network. The nodes in the sensor network cooperatively collect, process and route data towards the end user, the observer. An example of a sensor network is shown in Figure 2.2.

There is a wide range of applications for wireless sensor networks. In military operations sensor networks can be used for example to detect enemy movements, scout the terrain or locate the target area. The possible civil applications include habitat monitoring, weather observation and forecasting. Two examples of habitat monitoring prototype networks include the Great Duck Island project [17] to monitor seabird nesting and the PODS project [18] to monitor rare plants and their environment. In weather observation, sensors can be used to monitor temperature, rainfall and wind level in a remote terrain. This information can be used for example to forecast weather

and to detect tornados automatically.

Although wireless sensor networks require ad hoc networking of the nodes, there are differences between ad hoc and sensor networks. There are usually a lot more nodes in sensor networks than in ad hoc networks and therefore the network density is much higher. The energy conservation is a bigger issue in sensor networks because batteries have to be small and nodes are responsible for sensing and data processing in addition to communications. Thus, the topology changes caused by the on-off periods of the nodes are more frequent than in ad hoc networks. On the other hand, sensor nodes are usually more static than ad hoc nodes and the topology changes caused by mobility are rarer. The routing scalability in sensor networks is a bigger issue than in ad hoc networks because of the higher amount of nodes.

Among the three tasks of a sensor node, data transmission is much more expensive energy-wise than sensing or data processing. In [19], it was pointed out that it takes more than 100000 times less energy to execute a 32-bit instruction than to send 100 bits for 100 m using an RF link. Therefore, it is beneficial to aggregate and process data locally among a cluster of nodes as much as possible. Clusters are formed among neighbor nodes and a cluster head is elected to manage the group. Data aggregation and processing can be done locally inside a cluster by the cluster head and the cluster head can collectively transmit the summarized sensing data to other clusters towards the observer. Thus, as the number of links used in communications is reduced, hierarchical clustering decreases scalability problems in data dissemination.

Another way to conserve energy in sensor networks is to make them "too dense" by placing unnecessarily many sensors near the investigated phenomenon. This redundancy can be exploited to maximize the network lifetime by making only the minimum number of nodes do sensing while letting the other nodes save energy. The same strategy can also be used in sensor communications — only the minimum number of nodes that are needed to achieve a certain level of connectivity actively participate in routing.

Chapter 3

Medium access control in ad hoc networks

This chapter gives an overview of MAC protocols used in ad hoc networks. First to understand the contention for the wireless medium, the interference models for ad hoc networks are described. The ad hoc MAC protocols are classified into five groups: 1) traditional protocols used in packet radio networks, 2) protocols based on medium reservation, 3) power aware protocols, 4) protocols using directional antennas and 5) multiple channel protocols. The most important protocols from each of these groups are then presented. Finally, medium access control in sensor networks is considered.

3.1 Overview

In the IEEE 802 reference model, medium access is the function of the layer 2 Data Link sublayer called Medium Access Control layer. The main tasks of the MAC layer are to provide nodes with fair access to the medium and to avoid and resolve conflicts among the nodes. The MAC techniques are commonly divided into controlled channel access and random channel access. In controlled channel access techniques the medium is divided into non-overlapping channels that can be time slots (TDMA) or frequency bands (FDMA) or the medium is partitioned using orthogonal spreading codes (CDMA). When controlled channel access is used, a channel is usually allocated centrally to a single pair of nodes and there is no contention for the channel. In random channel access techniques the same channel is shared by multiple nodes that compete for the channel. A random access MAC protocol can be thought of as a distributed scheduling algorithm that randomly allocates the channel to

requesting nodes.

Because there is no central control in ad hoc networks, controlled channel access techniques are difficult to implement. Therefore, we will consider here only random access MAC protocols. To avoid interference between nodes sharing a channel, random access MAC protocols use control packets for channel reservation. The efficient use of control packets reduces the number of retransmissions caused by destructive interference (collisions) but on the other hand, the control traffic itself reduces the available bandwidth for data traffic. The traditional MAC protocols used in packet radio networks have no control traffic. Thus, collisions are common and the overall throughput is modest. Most MAC protocols used and proposed nowadays use control packets to reserve the channel for the transmission period. Depending on the protocol, the control packet exchange can reduce the collision probability dramatically with the cost of increased delay and increased node complexity.

The recent research on ad hoc MAC protocols has concentrated on either improving MAC performance by utilizing advanced hardware such as directional antennas and multichannel transceivers or fulfilling additional requirements such as energy efficiency and QoS. We will describe protocols belonging to each of these groups with the exception of QoS-aware MAC protocols which are not considered due to space limitations. For a review of selected QoS-aware MAC protocols, see [5].

3.2 Interference models

An ad hoc node sharing a common frequency channel with other ad hoc nodes hears transmissions from any node that is within its communication range. If there are two or more nodes transmitting simultaneously, the transmissions interfere each other. The amount of interference at the observer depends on the distances to the senders and the radio link properties. We now introduce the most common models to represent the communication range and the effect of interference.

The earliest model, the Boolean model, was already used in the research of packet radio networks. In the Boolean model it is assumed that each node has a common fixed transmission radius. Each node within the transmission radius of the sender hears the transmission and outside the transmission radius no node is able to hear the transmission. A node successfully receives the transmission if it hears no other transmissions (and is not transmitting itself). If there are two or more transmissions within the transmission radius of a node, a collision occurs and the node is not able to receive any transmissions.

The Boolean model was modified when packet radio nodes were equipped with capture receivers. When capture is used, the receiving node may be able to receive the transmission with the strongest signal. The power of the received signal is assumed to attenuate according to the power-law attenuation function that is given by

$$L(i,j) = Kl(i,j)^{\alpha}, \tag{3.1}$$

where K is an environmental constant and $l(i,j)$ is the distance between nodes i and j. It is commonly used to evaluate the average radio link path loss from point x to point y. When node j transmits to node i, the received power can be expressed as

$$P_{ji} = \frac{P_j}{L(i,j)} = \frac{P_j}{Kl(i,j)^{\alpha}}, \tag{3.2}$$

where P_j is the transmission power of node j and K and α satisfy $K > 0$, $\alpha > 2$. Thus, receiving node i is able to capture the signal of the nearest node if

$$\frac{P_{1i}}{P_{2i}} \geq \frac{1}{\beta} \qquad \text{or} \qquad \frac{l(i,2)}{l(i,1)} \geq \frac{1}{\beta^{1/\alpha}}, \tag{3.3}$$

where β is the capture ratio, P_{1i} and P_{2i} are the received powers of the nearest and the second nearest senders, and $l(i,1)$ and $l(i,2)$ are the corresponding distances.

The Physical Model [20] uses the power-law attenuation function to describe the effect of interference. Let $e_i \in 0,1$ indicate the state of node i such that $e_i = 1$ when i is transmitting and $e_i = 0$ when i is ready to receive. Now the transmission from node j to node i is successful when the Signal-to-Interference-and-Noise Ratio (SINR) exceeds a common threshold T,

$$\frac{P_j/L(i,j)}{N + \sum_{k \neq i,j} e_k P_k / L(i,k)} \geq T, \tag{3.4}$$

where N is the background ambient noise. Notice that the Physical Model takes into account the background noise and the amount of interference is the sum of all interfering signals. The capture model as characterized by (3.3) takes into account only the nearest interferer.

The power-law attenuation function is a deterministic function of distance. However, in reality the received power level fluctuates because of fading. Fading is a general term for power level fluctuations caused by signal reflection, refraction and diffraction from objects on the radio link path. Fading is divided into fast fading and slow fading. Fast fading is due to direct and scattered signals interfering each other. Slow fading is due to dominant shadowing objects on the radio link path. The power level fluctuation frequency is different in

slow and fast fading. When slow fading is modeled, the power level is considered constant during the transmission of a packet. In fast fading the changes in power level occur during the transmission of a packet.

The effect of slow fading is modeled in the shadowing model, which allows random power variations around the mean power. The path loss is a log-normal distributed random variable with a mean given by the power-law attenuation function. The path loss in dB is given by the sum,

$$\begin{aligned} L_2(i,j) &= 10\log(P_j/P_i) + X \\ &= 10\log L(i,j) + X = 10\alpha\log(Kl(i,j)) + X \quad \text{dB}, \end{aligned} \tag{3.5}$$

where X is normally distributed with zero mean and variance σ^2, $X \sim N(0,\sigma^2)$. The probability density function of X is denoted by $f_X(x)$. Let L_{th} be the threshold for the path loss such that $L_2(i,j)$ must be less or equal than L_{th} to make communication from node j to node i possible. As given in [21], the probability for a link between nodes i and j equals

$$\begin{aligned} \mathbf{P}[L_2(i,j) \le L_{\text{th}} \mid l(i,j)] &= \int_{-\infty}^{L_{th}-10\alpha\log(Kl(i,j))} f_X(x)dx \\ &= \frac{1}{2} - \frac{1}{2}\,\text{erf}\left(\frac{10\alpha}{\sqrt{2}\sigma}\log\frac{l(i,j)}{r_0}\;dB\right), \end{aligned} \tag{3.6}$$

where $r_0 = 10^{\frac{L_{th}}{\alpha 10\ \text{dB}}}$ is the maximum distance granting a link in the absence of slow fading ($r_0 = R$ in the Boolean model) when $K = 1$. The link probability between nodes as a function distance, when path loss exponent $\alpha = 3$, is shown in Figure 3.1.

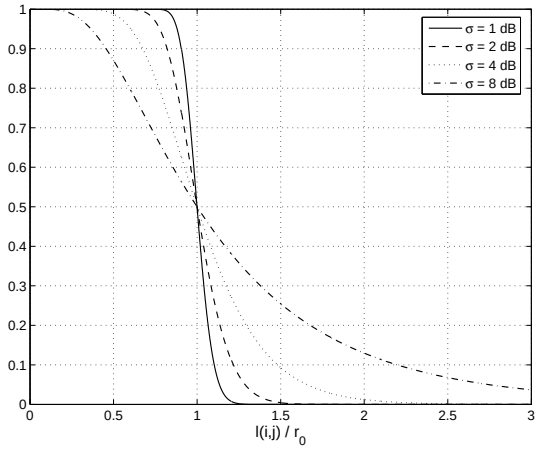


Figure 3.1: The probability that there is a link between a pair of nodes as a function of distance. The path loss exponent $\alpha = 3$ and $K = 1$.

The effect of fast fading can be modeled using the Rice or Rayleigh distributions for short-term power variations. However, these models are rarely considered when designing MAC or routing protocols because the effect of fast fading can be greatly reduced at the physical layer with advanced modulation and antenna techniques.

3.3 Medium access in early packet radio networks

ALOHA was the first random access MAC protocol for wireless networks [3]. It was designed for the full mesh single-hop ALOHA network, in which the only frequency channel was shared by all the nodes in the network. The idea behind ALOHA is extremely simple. Every node transmits a packet independent of any other node whenever data is available. If two or more transmissions take place at the same time, a collision occurs and no node is able to receive packets. Because the receiver acknowledges the succesful transmissions, the sender assumes that collision has happened if it does not receive an acknowledgement. In case of collision packets are retransmitted after a random waiting time to avoid synchronized collisions. If the transmission time of the packet is T and a node is to transmit a packet successfully, other nodes should not attempt to transmit within T before or after the start of the transmission. This $2T$ long vulnerable period is the key concept when analyzing the maximum throughput of ALOHA. Abramson [3] showed that under the Poisson traffic assumption the maximum throughput is $1/(2e) \approx 0.184$. Here the throughput means that the channel is used during 18.4 % of the time.

The slotted ALOHA was presented by Roberts [22] as an improvement of ALOHA. The slotted ALOHA assumes that the nodes are synchronized and the time is divided into time slots. The time slot size is matched to the fixed packet transmission time. The slotted Aloha allows transmissions to be started only at the beginning of a time slot. Now the collisions can occur only when two or more packets are transmitted at the same time slot reducing the vulnerable period to T. The acknowledgement procedure and random waiting times are similar to the basic ALOHA. Under the same Poisson traffic assumption the maximum throughput of the slotted ALOHA is $1/e \approx 0.368$. Because the vulnerable period is cut into half, also the maximum throughput is doubled. The major drawback of the slotted ALOHA is the need for synchronization between nodes. The perfect synchronization is difficult to maintain and as the clock drift between nodes increases, the performance of the slotted ALOHA

approaches that of ALOHA.

The most popular MAC protocol for early packet radio networks was the Carrier Sense Multiple Access (CSMA) presented by Kleinrock and Tobagi [23]. The idea in CSMA is to avoid collisions by sensing the channel before a transmission and transmitting only if the channel is idle. There are three variations to the CSMA protocol: non-persistent CSMA, 1-persistent CSMA and p-persistent CSMA. The operation of each of these variants is as follows:

- Non-persistent CSMA
  1. If the channel is sensed idle, transmit the packet immediately.
  2. If the channel is sensed busy, wait a random period of time and repeat channel sensing.
- 1-persistent CSMA
  1. If the channel is sensed idle, transmit the packet immediately.
  2. If the channel is sensed busy, continue sensing until the channel becomes idle and then transmit immediately.
- p-persistent CSMA
  1. If the channel is sensed idle, transmit the packet with probability p and wait for a time unit (the packet transmission time) with probability $1 - p$. Repeat channel sensing after the waiting period.
  2. If the channel is sensed busy, continue sensing until the channel becomes idle and then go to Step 1.

The 1-persistent CSMA is suitable only for light traffic situations. If there are two or more nodes waiting for the channel to become idle, they eventually transmit at the same time and the collision is certain. The best throughput is achieved with non-persistent and p-persistent CSMA with small p [23]. However, the delay can be long with non-persistent and p-persistent CSMA if the random backoff time is too long.

There are two fundamental problems related to CSMA and all other carrier sensing MAC protocols: the hidden-terminal problem [24] and the exposed terminal problem. The hidden terminal problem occurs in situations where there is a node transmitting within the communication range of the receiver but no other nodes transmitting within the sensing range of the sender. For simplicity, let us consider the case where the communication and sensing ranges

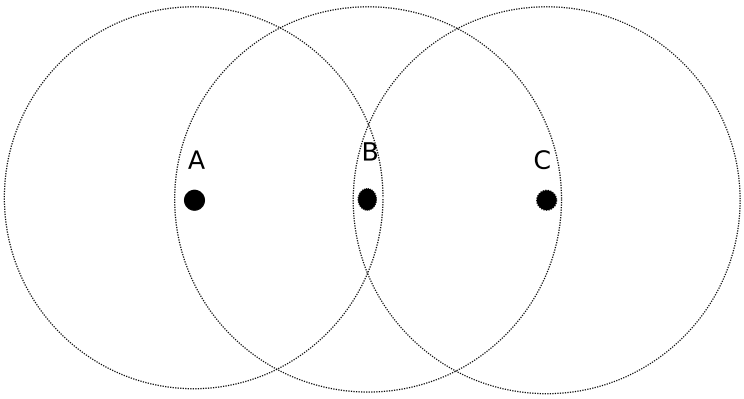


Figure 3.2: Hidden and exposed terminal problems. The sensing range equals the communication range that is depicted as a circle.

are equal as shown in Figure 3.2. Let us assume that Node A is transmitting to Node B. Because Node A is out of the sensing range of Node C, it is possible that Node C transmits simultaneously causing a collision at node B. Nodes A and C are hidden from each other. While the transmissions of hidden terminals cause collisions at the receiver, the hidden terminals should, however, be allowed to receive data packets.

The exposed terminal problem occurs when a node needlessly defers its transmission. In figure 3.2 if Node B transmits to Node A, Node C senses it and defers its transmission to any other node. This is unnecessary because C's transmission cannot cause a collision at Node A. Node C is said to be exposed to Node B.

3.4 Random access with medium reservation

Multiple Access with Collision Avoidance (MACA) [25] was proposed as an improvement over CSMA to eliminate the hidden and exposed terminal problems. MACA was one of the first random access protocols for wireless networks to use control packets for medium reservation. MACA introduced a control packet handshake between a sender and a receiver to ensure that neighboring nodes are aware of the upcoming transmission. If a sender wants to transmit a packet to a receiver, it sends an RTS (request to send) control packet to the receiver. The RTS packet includes the size of the data packet for estimating the duration of the data transmission. The receiver replies with a CTS (clear to send) packet that also includes the size of the data packet. After receiving the CTS packet, the sender can start transmitting the actual data packet.

The idea of the packet exchange in MACA is that a node can deduce from the control packets it hears when it is safe to transmit. Let us consider again the situation in Figure 3.2 when Node A is transmitting to Node B. Node C cannot hear the RTS packet sent by Node A but it hears the CTS packet sent by Node B. From this it can deduce that it is within the range of a receiver and also how long the data transmission is expected to last. Now Node C defers its own transmissions for a period that is a sum of the expected length of the data transmission between A and B and a random backoff time. If Node B is transmitting to node A, Node C hears the RTS packet sent by Node B but not the CTS packet sent by Node A. Now after hearing the RTS packet, Node C must wait for a short period during which it listens for a CTS packet. If it does not hear the CTS packet, it deduces that it is out of the range of the receiver. After that Node C is allowed to send RTS to a chosen receiver (other than B) after a random backoff time.

MACA substantially reduces the collision probability compared to CSMA by replacing carrier sensing with the control packet exchange. However, collisions between data packets can still occur if there are collisions among control packets. In the worst case control packet collisions can effectively ruin the channel reservation and the performance of MACA degenerates to that of ALOHA. Usually the control packets are significantly shorter than data packets and collisions are less likely.

MACA does not provide acknowledgement for data packets and in the case of data transmission failure, the retransmission has to be initiated by an upper layer protocol. This would increase the overall transmission delay. MACA for Wireless (MACAW) [26] was presented by Bharghavan et al. to increase the reliability of MACA by acknowledging a data packet with an ACK control packet and to increase the fairness of the random backoff scheme. Bharghavan et al. pointed out that it is useless for an exposed terminal to initiate transmission during a transmission going on within its range because it is unable to receive the CTS packet. They also stated that the transmission from an exposed terminal can collide with ACK packets within its range and therefore exposed terminals should wait until the end of the ongoing transmission before starting their own transmission.

MACAW uses a five step RTS–CTS–DS–DATA–ACK packet exchange. Hidden terminals defer from transmitting until the end of the transmission when they hear a CTS packet and exposed terminal defer transmitting when they hear a DS (Data-Send) packet. The purpose of the DS packet is to inform the exposed nodes that the RTS–CTS exchange was successful and not to make exposed nodes defer their transmissions in vain. This more complex control exchange results in better performance than MACA when the channel load is

high.

In order to guarantee collision-free data transmission, Floor Acquisition Multiple Access (FAMA) [27] combines carrier sensing with the control packet exchange. In FAMA the sender uses non-persistent carrier sensing scheme described in Section 3.3 to make sure the channel is idle before transmitting an RTS packet. The receiver responds with a CTS packet if no transmissions are taking place within its range. Because no other nodes start transmitting after hearing an RTS or a CTS packet, this procedure should acquire the channel (the floor) for the data transmission.

IEEE 802.11 MAC

The IEEE 802.11 [28] is a wireless LAN standard defining the physical layer specifications and the IEEE 802.11 MAC protocol. The IEEE 802.11 MAC protocol includes two variants: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). PCF is a centralized MAC scheme used to provide contention-free service. PCF is suitable for a wireless LAN that has some sort of base station coordinating the access to the channel. DCF uses a contention algorithm based on channel reservation in a distributed fashion. Because PCF cannot be used in ad hoc networks, we concentrate on describing the DCF scheme of the IEEE 802.11 MAC protocol.

DCF has the same basic idea as FAMA of combining carrier sensing with the control packet exchange in order to minimize the collision probability. In addition, DCF includes short additional delays before sending a packet. A short interframe space (SIFS) is used before sending a response packet such as CTS or ACK and a longer distributed coordination function interframe space (DIFS) is used before sending a data or RTS packet. A node that has a packet to transmit operates as follows:

1. Sense the channel. If the channel is idle, wait for a DIFS. If the channel is still idle, start transmitting (RTS-CTS-DATA-ACK exchange).

2. If the channel is busy, defer transmitting and continue sensing the channel.

3. When the transmission that had reserved the channel is over, wait for a DIFS. If the channel is idle, wait for a random backoff time (use the frozen backoff time if it exists). If the channel is still idle, start transmitting. If the medium becomes busy during the backoff time, freeze the backoff time and go to Step 2.

4. If no CTS or ACK was received, collision has occurred. Go to Step 1.

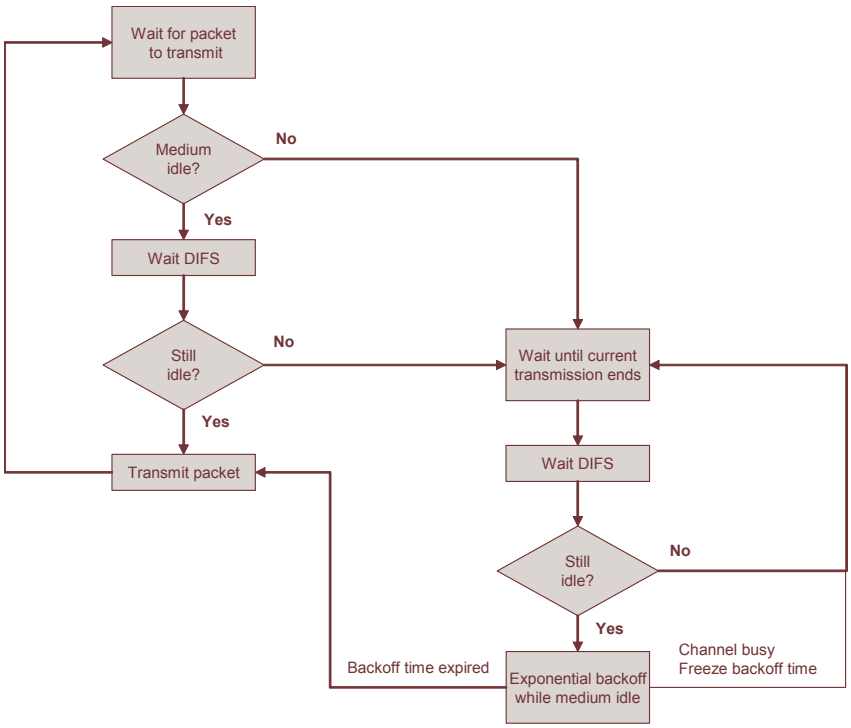


Figure 3.3: IEEE 802.11 DCF medium access control logic [29].

The access control logic of DCF is also shown as a flow chart in Figure 3.3.

The purpose for the use of two interframe spaces of different length is to provide priority to reactive traffic (CTS and ACK packets). For example, consider two nodes A and B initiating a transmission and node C that attempts to start transmitting a little later. Node B attempts to send a CTS packet in response to an RTS packet from node A and at the same time, Node C attempts to initiate a new transmission by sending an RTS packet. Now because B has to wait only for a SIFS and C has to wait for a DIFS, B starts transmitting its CTS packet first and C has to back off. This is illustrated in Figure 3.4. Generally, any node using SIFS has the highest priority, because it will always gain access prior to a node waiting for a time equal to DIFS.

DCF uses the binary exponential backoff algorithm to vary the random backoff time. When a transmitting node notices that a collision has occurred, it doubles its mean value of the backoff time before attempting to retransmit. Repeated failed transmission attempts result in longer and longer backoff times until the maximum value is reached. The mean of the backoff time is restored to its minimum whenever a node successfully completes a data transmission. This kind of exponential increase in backoff times helps to smooth out the heavy load. On the other hand, if the backoff times of the nodes are

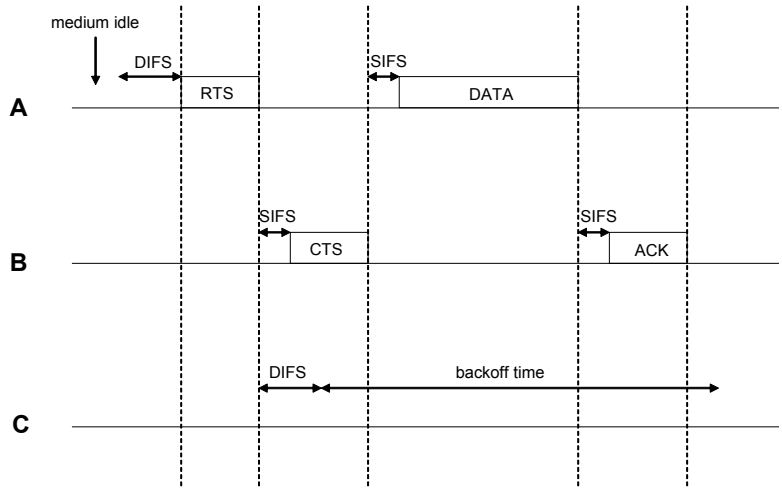


Figure 3.4: Impact of different interframe spaces.

significantly different, the binary exponential backoff algorithm can result in unfairness.

The IEEE 802.11 is the most used standard in wireless LANs but it was not particularly designed for multihop networks. Xu and Saadawi [30] simulated the IEEE 802.11-based multihop ad hoc network with TCP traffic to find out problems that are related to the multihop nature of the network. One of the problems they obrserved was the unfairness caused by the binary exponential backoff scheme. Because the mean of the random backoff time is doubled in the case of a collision, the scheme always favors nodes whose last transmission attempt was successful.

Although all reservation-based MAC protocols presented in this section are designed to solve hidden and exposed terminal problems, none of them completely succeeds in this task. It was pointed out in [31] that even if an exposed node is allowed to send their RTS packets to another node, the CTS response will collide with the on-going data transmission and the exposed node will not be able to start transmitting data. Also, even if a hidden node is allowed to receive, it cannot reply to an RTS packet with CTS because of the risk of collision. This is illustrated in Figure 3.5. Exposed node C cannot send data to node E because it does not hear the CTS packet. Correspondingly, hidden node D cannot receive data from node F because it cannot reply with a CTS packet.

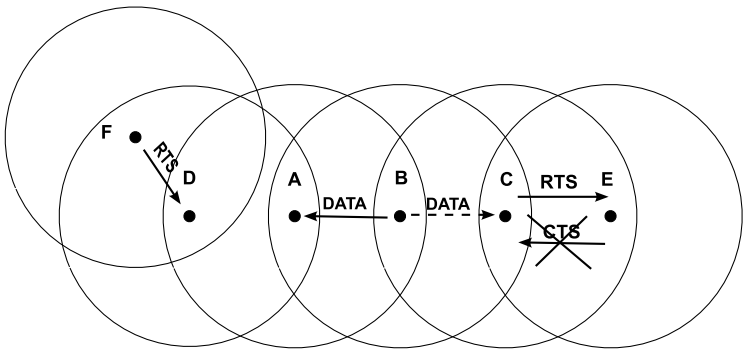


Figure 3.5: Hidden and exposed node problems related to single-channel reservation-based MAC prototocols.

3.5 Power aware MAC protocols

In principle, there are three ways to reduce energy consumption at the MAC layer:

1. Avoid collisions. As the retransmissions consume energy, the collision probability should be kept as low as possible.

2. Power management. Allow nodes to turn on the standby mode or switch off during idle times. Especially carrier sensing consumes needlessly energy when there are no transmissions going on.

3. Power control. Adjust the transmit power level according to the location of the receiver. Power control can not only save energy but also increase spatial channel reuse and thus the capacity in the network.

Since the collision avoidance in some form is included in all MAC protocols, we present here protocols that achieve energy savings using power management or power control.

Singh and Raghavendra [32] pointed out that ad hoc nodes consume a considerable amount of energy also when they are receiving transmissions. Due to the broadcast nature of wireless ad hoc networking, there are nodes that consume energy while overhearing the transmissions not directed to them. Their suggestion to this problem was Power Aware Multi-Access protocol with Signalling (PAMAS) that makes nodes to power off while not actively transmitting or receiving packets.

PAMAS is based on the MACA protocol with the addition of a separate control channel. Like in MACA both RTS and CTS packets include the expected length of the upcoming data transmission but in PAMAS all control packet

exchange takes place in the control channel. There are two cases when PAMAS makes a node to power off. If the node has no packets in its queue and its neighbor starts transmitting, it powers off for the expected duration of the transmission. Also when at least one neighbor is transmitting and another is receiving, the node powers off for the expected transmission duration even if it has packets to send. If the node powers back on and there are on-going transmissions, it has to start a probe packet exchange in the control channel to find out the length of the next power-off period. The power-off rules have no effect on the throughput because nodes are powered off only in situations when they can neither send nor receive packets.

The Power Controlled Multiple Access (PCMA) protocol [33] was primarily designed to improve the spatial reuse in an ad hoc network using power control and the power saving was only the secondary goal. The idea behind PCMA is to use the control channel to advertise the maximum interference power that receivers can tolerate. The transmission powers of new senders are then tuned according to the acceptable power level. The operation of the sending node i and the receiving node j is as follows:

1. Node i listens to the busy tone pulses in the control channel to find out what is the maximum allowed transmit power P_t. When P_t is greater than the pre-defined minimum transmit level P_{min}, i waits a random backoff time. If $P_t > P_{min}$ during the whole backoff time, i sends a Request Power to Send (RPTS) packet to Node j with the power P_t using the data channel. The P_t value is included in the RPTS packet.

2. When j receives the RPTS packet, it uses the received power level and the P_t value to calculate the minimum required power level P_r for i's transmissions. The value of P_r is included in the Acceptable Power to Send (APTS) packet. Node j then responds to i with the APTS packet that is sent with the maximum allowed transmit power level at j.

3. When i receives the APTS packet, it checks if $P_r < P_t$. If the condition is fulfilled, it sends the data packet with the power P_r. In the case that $P_r \geq P_t$ or the APTS is not received at all, i increases its backoff time and goes to Step 1.

4. As soon as i starts the reception of data packet, it start sending periodic busy tone pulses to the control channel. Potential interferers listen to the control channel and can deduce from these pulses their maximum allowed transmit powers.

5. When i has received the whole data packet, it sends an ACK packet back to j.

6. If i receives the ACK packet, it resets its backoff time and goes to Step 1. If ACK is not received, the backoff time is increased before the retransmission.

PCMA clearly improves the channel utilization as the spatial reuse is improved and also the power used in transmissions is reduced. The cost for this is the increased node complexity due to power level measurements and the use of a distinct control channel. In addition, collisions in the data channel are still possible. The collision probability increases in heavy traffic situations because the busy tone pulses collide more frequently resulting in miscalculations of allowed transmit powers.

The IEEE 802.11 DCF mode [28] also supports power management. In the DCF power saving mechanism each node is either in the active mode or in the power saving (PS) mode. Nodes in the PS mode wake up only for the duration of the Ad-hoc Traffic Indication Message (ATIM) window to check whether there are data packets directed to them. In the beginning of the ATIM window there is a beacon message exchange in order to synchronize the nodes. After the synchronization is complete, each node that has packets to send transmits an ATIM packet that is acknowledged by the receiver. If a node does not send the ATIM or ACK packet, it has no data to transmit or receive during the data exchange. The idle nodes can set themselves to the standby mode until the beginning of the next ATIM window.

The power saving mechanism of the IEEE 802.11 DCF mode is not suitable for multi-hop networks because it requires that all nodes are synchronized and fully connected. In a multi-hop mobile ad hoc network synchronization is difficult because communication delays and node mobility are unpredictable. Even if perfect synchronization is available, it is possible that node mobility causes the network to become partitioned into clusters each having independent synchronization.

Some protocols have been proposed to include power control to the IEEE 802.11 MAC protocol. The basic idea of these protocols is that the RTS-CTS exchange is done with the maximum power level so that neighbors become aware of the upcoming transmission. The DATA-ACK exchange can then be done with the minimum required power level. Jung and Vaidya [34] pointed out that this kind of power control is harmful if the transceivers are sensitive to interference. For example, if there is a node that can sense the CTS packet sent by the receiver but is not able to decode the CTS packet correctly, it defers transmitting for the extended interframe space (EIFS) period. After the EIFS period the node is not able to sense the data packet sent with a lower power. Thus, the node can start transmitting an RTS packet with the full power possibly causing interference to the ongoing data transmission. As a solution,

Jung and Vaidya proposed that the sender periodically raises the power level to the maximum to keep neighbors aware of the ongoing transmission.

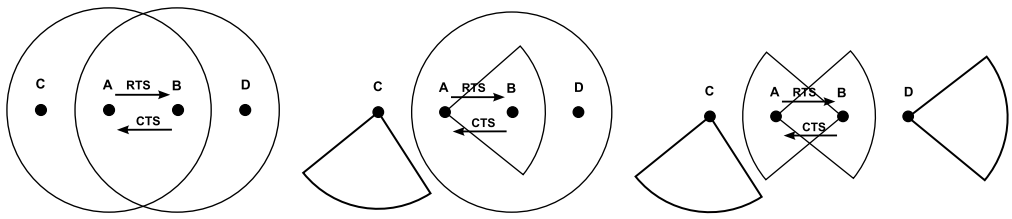
3.6 Medium access control using directional antennas

The MAC protocols considered thus far assume the use of omni-directional antennas that can transmit signals to and receive signals from all directions. Directional antennas have a directional radiation pattern that makes it possible to transmit only to the nodes in the wanted direction. The benefit of the directional antennas is that the interference to the nodes outside the directional pattern is significantly reduced. Similarly, if the receiver only listens to the direction of the sender, the received interference signal power from other directions is negligible. For these reasons, it is clear that the use of directional antennas leads to a better spatial reuse.

Directional antenna systems can be coarsely divided into three groups: sectorized antenna systems, directional beam-forming systems and multi-beam adaptive array systems. A sectorized antenna system consists of M directional antennas each having a conical radiation pattern with an angle of $2\pi/M$ radians. For the directional transmission or reception only one of the directional antennas is used simultaneously. A directional beam-forming system has a single antenna that is able to form a beam towards the receiver or sender. This is done by adaptively steering the antenna towards the desired direction. A multi-beam adaptive array system (smart antenna system) is able to form multiple beams for several simultaneous receptions or transmissions.

It is clear that directional antenna communications require more complex and expensive hardware than omni-directional antenna communications. Therefore, directional antennas are not suitable for all ad hoc networking applications. We consider here protocols for antenna systems that are able to receive or send only one transmission simultaneously.

There have been several proposals to add directional antenna support to the IEEE 802.11 MAC protocol. It is common to all these proposals that data packets are transmitted using directional antennas. The difference between these proposals is whether RTS and CTS packets are transmitted directionally or omni-directionally. The schemes covered here are oRTS/oCTS using both omni-directional RTS and omni-directional CTS packets, dRTS/oCTS using directional RTS and omni-directional CTS packets and dRTS/dCTS using both directional RTS and directional CTS packets. For an illustration of these schemes, see Figure 3.6.



(a) oRTS/oCTS (b) dRTS/oCTS (c) dRTS/dCTS

Figure 3.6: The control packet exchange schemes based on the use of directional antennas

Nasipuri et al. [35] presented a protocol based on the oRTS/oCTS scheme with the use of sectorized antennas. When a node receives an oRTS packet addressed to itself, it checks which antenna received the maximum power to find out the direction of the sender. Similarly, the sender can deduce from the received oCTS packet the direction of the receiver. Thus, the data transmission can be done using directional antennas without prior knowledge of the location of the receiver. Note that the oRTS/oCTS scheme does not increase spatial reuse compared to the basic IEEE 802.11 MAC protocol because all nodes hearing oRTS or oCTS defer transmitting. However, the background interference level at the nodes is decreased because data is transmitted using directional antennas.

Ko et al. [36] proposed a dRTS/oCTS protocol as a tradeoff between the increased spatial reuse and the collision probability. As it can be seen from Figure 3.6b, the exposed node C is now allowed to transmit simultaneously with A. This results in the increased spatial reuse but also collisions at A are possible if C's destination happens to be in the same direction as A. Wang and Garcia-Luna-Aceves [37] showed that the more aggressive dRTS/dCTS scheme results in better throughput and less delay than the dRTS/oCTS scheme. In their proposed protocol both hidden and exposed nodes (C and D in Figure 3.6c) are allowed to transmit. The spatial reuse in their dRTS/dCTS protocol is excellent but because the collisions are more common, the backoff algorithm should be carefully designed. The drawback of the dRTS/oCTS and the dRTS/dCTS protocols is that they require the location information of the receiver before transmission.

3.7 Multiple channel random access

Basically, multiple channels can be exploited in the random medium access in two different ways. The first approach is to separate the medium into the data

channel and the control channel. The control channel is used for reserving the medium in such a way that the hidden and exposed terminal problems are eliminated. The second approach is to use multiple channels for data transmission. Because multiple channels can be used simultaneously within the coverage area of a node, the throughput is increased and the collision probability is decreased.

The dynamic and distributed assignment of multiple channels to different nodes is a difficult issue. Most of the proposed protocols in the literature assume perfect synchronization among the nodes. As mentioned earlier, the synchronization in a multihop environment is challenging because of unpredictable delays and node mobility. In practise, synchronization with a given accuracy can be achieved by using the Global Positioning System (GPS). Since the clock synchronization issues are out of the scope of the study, we consider only multiple channel random MAC protocols that require no synchronization.

One of the most cited dual channel ad hoc MAC protocol is the Dual Busy Tone Multiple Access (DBTMA) protocol presented by Haas and Deng [31]. The idea of DBTMA is to send busy tones in the control channel to protect the RTS and data packet transmissions in the data channel. The transmit busy tone (TBT) is sent during the transmission of an RTS packet by the sender and the receive busy tone (RBT) is sent during the data transmission by the receiver. Each node hearing a busy tone must defer transmitting. The operation of DBTMA is as follows:

1. If a node has a packet to send, it starts sensing the control channel. If the channel is idle, it sends both an RTS using the data channel and a TBT using the control channel to the receiver. If the channel is busy, the node draws a random backoff time and senses again.

2. When the receiver receives the RTS packet, it starts sending an RBT.

3. When the sender senses the RBT from the receiver, it starts transmitting the data packet.

4. When the receiver has received the whole data packet or the transmission timer has expired, it stops sending the RBT.

The advantage of using a distinct control channel compared to single-channel reservation-based MAC protocols is that the exposed and hidden terminal problems presented in Section 3.4 are solved. The hidden nodes cannot start transmitting as long as their hear a RBT signal but they can receive since RBT signals are sent using the control channel. The exposed nodes can start transmitting (after backoff time) as soon as they hear no TBT signal. However,

there are a couple of drawbacks related to the DBTMA protocol. First, the nodes need to be equipped with hardware that allows the transmission of the RTS packet and the TBT signal simultanously. Second, because there are no acknowledgement packets, upper-layer protocols are responsible for the possible retransmissions.

Since DBTMA both increases power consumption by continuously sending busy tones and decreases the number of retransmissions, it should be interesting to find out whether DBTMA is more effective energy-wise than single-channel reservation-based protocols. Note that the use of distinct control channel is also utilized in PAMAS [32] and PCMA [33] to reduce the energy consumption, as was discussed in Section 3.5.

The multichannel CSMA protocol presented by Nasipuri et al. [38] assumes that the frequency range is divided into N channels using either FDMA or CDMA techniques. The transceiver at a node should be able to sense all N channels simultaneously and transmit or receive using any of these channels at a time. The operation of the multichannel CSMA protocol is similar to the non-persistent CSMA with the addition of the channel selection algorithm. The sender first senses all channels to find out which of these are idle. If the channel that was used for the last successful transmission is idle, it is selected. Otherwise, one of the free channels is chosen at random. In the case that all channels are busy, the sender waits for a random backoff period before resensing.

Because the multichannel CSMA reduces collisions, it increases the network throughput in most cases when compared to the single-channel CSMA. However, because the channel splitting does not increase the overall capacity of the medium, the performance of the multichannel CSMA is lower than that of the single-channel CSMA in light traffic situations.

Another multichannel MAC protocol is the Dynamic Channel Assignment (DCA) protocol [39] that uses a control channel for all control traffic in addition to N data channels. Nodes are required to have two transceivers, one to operate on the control channel and the other to dynamically switch to one of the data channels. Each node has two tables that it updates dynamically: the Channel Usage List (CUL) and the Free Channel List (FCL). Whenever a node receives a control packet, it checks which node is active and in which channel. This information and the expected duration of the channel usage are updated dynamically into CUL and FCL is dynamically computed from CUL.

When the sender wants to communicate with the receiver, it sends an RTS packet with its FCL information to the receiver. The receiver compares this FCL with its CUL to find out if there are any idle channels to use. The receiver responds with a CTS packet including information about which channel is

selected for the upcoming data transmission. The other nodes within the range of the receiver (hidden nodes) can then defer using that channel and update their CULs. When the sender receives the CTS packet, it sends a reservation (RES) packet including the reserved channel ID. Correspondingly, all nodes that hear the RES packet (exposed nodes) defer using the reserved channel and update their CULs. Now the data packet can be transmitted using the reserved channel and the successful reception of the data packet is acknowledged normally. When there are no free channels for transmission or a collision has occurred, the sender must back off for a random time.

DCA was later improved to support power control. In DCA with Power Control (DCA-PC) [40], RTS and CTS packets are sent using the maximum power level and data and ACK packets are sent using the minimum power level that enables communication with the sender and the receiver. The receiver can deduce from the received power level of the RTS packet what the minimum required power level is. This information is included in the CTS packet. In addition to the CUL and FCL tables nodes maintain dynamically updated POWER tables. The POWER table has an entry for each neighbor about the power level that should be used when sending data to that neighbor. The power level information is collected from all RTS and CTS packets that are heard by the node.

Power control allows two pairs of nearby nodes to use the same channel if their used power levels do not interfere the transmission of the other pair. Information about maximum allowed power levels is available in POWER tables. Because of the better spatial reuse, DCA-PC achieves better throughput than DCA. However, it was shown that as the number of channels increase, the impact of power control becomes less significant.

3.8 Medium access control in sensor networks

Due to the miniature size of sensor nodes, their battery capacity is usually very limited. Each node consumes its energy resources in three tasks: sensing, data processing and communication with other nodes. The communication is the most energy-consuming of these tasks and therefore the primary goal of a sensor network MAC protocol is to operate in an energy-conserving manner. The typical performance measures used in wireless ad hoc networks, such as delay and throughput, are usually not equally important in sensor networks. Also, since all sensor nodes share a common task, the fairness among nodes is rarely a design goal for MAC protocols.

The research on sensor network MAC protocols is still in its infancy. Although

there are various proposed protocols, no protocol is accepted as a standard. It should be noted that there are also no standards for the lower physical layer and for the sensing hardware. One of the reasons for this lack of standards is that sensor networking, in general, is application-dependent. Thus, the proper choice for the MAC protocol depends on the investigated phenomenon, the density of the network etc. We will present here the Sensor-MAC (S-MAC) protocol [41] and its further improvement Dynamic Sensor-MAC (DSMAC) [42]. The primary function of these protocols is to control active-sleep cycles of the nodes in an energy-efficient way.

The basic idea behind S-MAC is to form clusters locally from neighboring nodes. The nodes within a cluster are kept synchronized with periodic synchronization (SYNC) packets in order to maintain the same sleep periods. The contention for the access to the channel is done similarly as in the IEEE 802.11 DCF with carrier sensing and the RTS-CTS-DATA-ACK exchange. Broadcast packets are sent without using RTS/CTS. We start considering the operation of S-MAC by introducing the synchronization algorithm.

S-MAC divides time into slots and each slot into two periods: listen and sleep. The listen interval length is fixed according to physical and MAC layer parameters. Each node maintains a schedule table containing listen-sleep schedules for all its neighbors. Initially all nodes are synchronized having the same schedule. The schedule tables are periodically updated to avoid clock drifts using SYNC packet exchanges. The SYNC packets include the id of the sender and the time of its next sleep relative to the moment when the sender starts transmitting the SYNC packet. When a receiver gets the SYNC packet, it can subtract the packet transmission time to find out the sleep time of the sender and adjust its timer accordingly. The SYNC packets are exchanged as follows:

1. A node listens to the channel for a fixed amount of time. If the node has not received a SYNC packet by a random backoff time, it chooses its own schedule, starts to follow it and broadcasts a SYNC packet to all of its neighbors.

2. If the node receives a SYNC packet from a neighbor before sending its own schedule, it sets its own schedule according to the received SYNC packet.

3. If the node receives a SYNC packet including a different schedule after it has chosen a schedule, it adopts both schedules by waking up at listen intervals of both schedules.

It should be noted from the above algorithm that the nodes belonging to two clusters consume more energy than others because they have to listen for two

intervals. The listen intervals are divided into two parts, the synchronization interval described above and the data interval. During the data interval a node can start transmitting an RTS packet if the channel has been idle until the end of its random backoff time. If the node hears a packet that is not directed to it, it can immediately switch to the sleep mode.

The other important feature in S-MAC is the reduction of control traffic by sending multiple packets using the same RTS-CTS exchange. Although only one RTS and one CTS packet are used to reserve the channel for a burst of packets, each data packet is individually acknowledged. The RTS and CTS packets as well as all data and ACK packets contain the expected length of the whole transmission. In the case of retransmissions, the expected length is updated to the remaining packets. Now any node overhearing any packet of the burst can go to the sleep mode until the end of the transmission.

S-MAC can clearly reduce the energy consumption significantly. This is done at the cost of decreased throughput, increased latency and unfairness. Throughput is reduced because only part of the time can be used for communication. Latency increases especially in situations where a message-generating event occurs during the sleep interval. Finally, the option to reserve the channel for a burst of packets results in unfairness.

DSMAC was proposed as an improvement on S-MAC to reduce the latency for delay-sensitive applications. The duty cycle is defined as the ratio of the listen interval length to the time slot length. DSMAC is able to dynamically double or halve the duty cycle according to the current traffic condition. Each node keeps track of the average latency of its packets and the energy consumption per a delivered packet. Here the latency is defined as the difference between the moment when a packet gets into the queue and the moment when the packet is successfully transmitted. The latency value is included in each data packet. When a receiver notices that the average latency is high and the energy consumption does not exceed its limit, it shortens its sleep interval length such that the duty cycle is doubled. The duty cycle is halved when the queue is empty or the average latency is smaller than a predefined limit.

The sleep interval lengths are included in SYNC packets. Whenever a node receives a SYNC packet with a shorter sleep interval length from its neighbor, it checks if it has packets directed to that neighbor. If there are such packets and its energy consumption does not exceed the limit, it shortens its sleep interval length to match its neighbor. It should be noted that because the increased duty cycle is always a multiple of the basic duty cycle, the listen intervals between neighbors are synchronized regardless of the duty cycle. This is illustrated in Figure 3.7.

DSMAC increases overhead compared to S-MAC since nodes have to keep

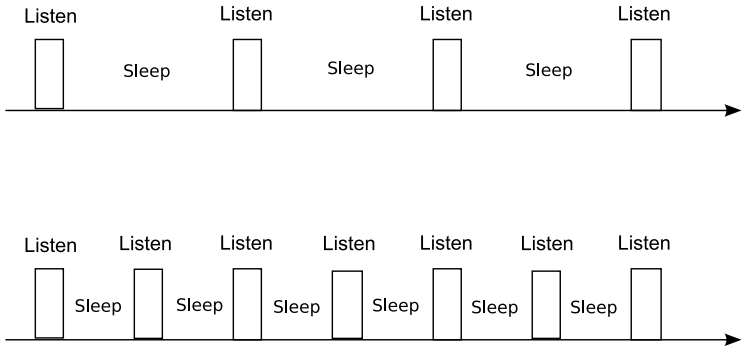


Figure 3.7: Two different duty cycles with listen intervals synchronized.

track of the average latency and the energy consumption. Also the overall energy consumption increases because of shorter sleep intervals. However, it was shown that DSMAC decreases the latency and the energy consumption per packet.

Chapter 4

Routing in ad hoc networks

This chapter gives an overview of routing techniques and protocols used in ad hoc networks. First we present broadcasting techniques used in route discovery. We make the common classification of ad hoc routing protocols into proactive and reactive and describe example protocols belonging to both of these groups. Then, geographic routing methods and location service protocols related to them are discussed. Finally, routing and topology control protocols for wireless sensor networks are considered.

4.1 Overview

Node mobility, shared wireless channel and energy limitations make routing in ad hoc networks very different from the traditional routing in wired networks. The movement and the power saving periods of the nodes result in rapid topological changes in the network topology. Routing protocols should be able to dynamically adapt to the changes in a fully distributed fashion. The bandwidth in the shared channel is a limited resource and it must be used as effectively as possible. Thus, the routing overhead should be as minimal as possible. Smaller routing overhead increases also energy-efficiency because more available battery power is used in payload data transmissions.

Ad hoc routing protocols have been traditionally divided into two main categories: proactive (table-driven) protocols and reactive (on-demand) protocols. Proactive routing protocols maintain tables at nodes to store routes between all pairs of nodes. Whenever a change occurs in the network topology, the routing tables need to be updated. Reactive protocols do not attempt to maintain the up-to-date topology of the network. When a route between the source and the destination is needed, a reactive protocol initiates a procedure

to find a route to the destination. Between the proactive and the reactive approaches there are hybrid protocols that hierarchically utilize both proactive and reactive protocols.

In addition to the proactive and the reactive protocols, we discuss routing based on geographical information, and routing in sensor networks. Broadcasting of routing messages is an essential part of many routing protocols based on route discovery. That is why broadcasting techniques are also considered before the actual routing protocols. The discussion in this thesis is restricted to unicast routing. Multicast routing in ad hoc networks has also been actively researched recently. For a comparison of a wide range of ad hoc multicast protocols, see [43].

4.2 Broadcasting techniques

When ad hoc nodes share a common wireless channel, each node hears transmissions from every node within its communication range. Thus, every transmission from a node can be considered as a broadcast to all its neighbors. Network-wide broadcasting can be easily achieved by letting each node, which receives a packet for the first time, further broadcast the packet to all its neighbors. This broadcasting technique is commonly called flooding. Flooding delivers the data from the source to every node that belong to the same connected part of the network as the source.

Flooding is not a very efficient way of delivering messages. The total number of transmissions needed to deliver a single packet from a source to a destination is in the order of the number of nodes in the network. Another inefficiency is that a node may receive the same packet multiple times. It is likely that every node in the network could be reached with fewer transmissions and fewer collisions. For these reasons, the simple flooding technique is not used for delivering data packets in ad hoc networks.

The advantage of the flooding technique is that no topological information must be known in advance. Thus, flooding is useful in route discovery of the reactive routing protocols. In order to improve the performance of the reactive protocols, research efforts have been made to modify the simple flooding technique. The purpose of these modifications is to reduce the number of redundant transmissions while still ensuring that all nodes receive the broadcasted packet.

The easiest way to modify the simple flooding technique is to make each node forward the packet with some probability. This broadcasting technique is called gossiping. Haas et al. showed that gossiping can reduce traffic up to

35 % when compared to the simple flooding [44]. The choice of the proper gossiping probability is important because with a too small probability the flood dies out before reaching every node and with a too large probability there are many redundant transmissions.

Another approach to reduce the redundant transmissions in flooding is to select only a subset of nodes for broadcasting. This forwarding set should be as small as possible still guaranteeing that every node receives the transmitted packet. The methods that heuristically search for a small forwarding set are called neighborhood-knowledge methods. Many of the neighborhood-knowledge methods need the knowledge of the neighbors within 2-hop radius to prune redundant nodes from the forwarding set. Different methods for efficient broadcasting are compared in [45]. It was shown that the neighborhood-knowledge techniques perform better than gossiping with the cost of higher overhead.

4.3 Proactive routing

The proactive routing protocols for ad hoc networks are divided into distance vector protocols and link state protocols. In distance vector protocols, routing traffic consists of distance vectors that are exchanged between neighboring nodes. A distance vector sent by a node contains distances from the node to all known destinations. The distance information is used to calculate a routing table at each node. The routing table entry typically has the distance to the destination and the next-hop node towards the destination. In link state protocols, each node broadcasts its link state information. Nodes use the link state information received from other nodes to form knowledge of the complete network topology. The topology information is usually stored in a tree structure since the source only needs to know the state of the links in the shortest path to the destination.

When a proactive routing protocol is used, nodes maintain routes to every other node in the network. This causes redundant route calculations because there may be routes that are rarely or never used. On the other hand, proactive routing reduces delay in data transmissions because the route is already available and no on-demand route discovery needs to be done.

4.3.1 Distance Vector Protocols

Destination-Sequenced Distance-Vector (DSDV) [4] was one of the first routing protocols primary designed for ad hoc networks. DSDV adds the use of

sequence numbers to the well-known distributed Bellman-Ford algorithm [46] in order to prevent routing loops. In addition to a routing table, each node maintains a monotonically increasing sequence number for itself. A node increases its sequence number before sending an update message. The routing table contains the following:

- list of all available destinations,
- next hop for each destination,
- number of hops to each available destination, and
- sequence number for each route table entry originated by the destination.

Each node broadcasts distance vectors periodically or after it notices significant topological changes. The periodic updates contain the number of hops to each destination and the event-driven updates contain only the hop counts to those destinations whose hop counts have changed since the last full update. All distance vectors contain also destination-originated sequence numbers for each destination entry and a common sequence number set by the sender. The receiver updates its routing table entry according to the distance vector if the destination-originated sequence number is higher in the distance vector. If the distance vector and the routing table entry have the same sequence number, the one with the smaller number of hops is used. The sequence number, which is included by the sender of the distance vector, is used to estimate the route settling time (the time between the arrival of the first and the best route for each destination). This information is stored in another table and is used to delay the broadcasts for destinations with a long route settling time. The purpose of the delays is to avoid advertising routes that may not have stabilized yet.

DSDV introduces large amounts of routing overhead to the network because of periodic routing table updates. This results in scalability problems in a large network because a significant fraction of available bandwidth is used in relaying the distance vectors. Therefore, DSDV is applicable only to small, relatively static ad hoc networks.

4.3.2 Link State Protocols

Link state protocols maintain a link state database of the whole network in each node. The link states can be stored, for example, in an adjacency matrix $\mathbf{A}$ where a nonzero entry a_{ij} means that there exists a link between nodes i and j. The routing table is generated from the link state database typically

using Dijkstra's shortest path algorithm [47]. Instead of storing the whole shortest path tree for each destination, it is enough to store only the next hop and the number of hops for each destination. The maintenance of link state databases is common to all link state routing protocols but there are differences in how the link state updates are arranged. We will consider here two common variants: Fisheye State Routing (FSR) [48] and Optimized Link State Routing (OLSR) [49].

The idea of FSR is to maintain more accurate routing information about the immediate neighborhood of a node while the routing information about far-away destinations can be less accurate. In FSR, link state packets are not flooded but each node sends a link state update only to its neighbors. A node receiving a link state update, changes its link state database and routing table entries only if the sequence number of the update is higher than the sequence number of the routing table entry. In order to reduce control traffic overhead, only periodic updates, instead of event-driven updates, are used. The link state updates are sent using different exchange periods depending on the hop distance for the routing table entry. The idea is to send link state updates corresponding to the near links more frequently than those corresponding to the links that are multiple hops away. For example, each update sent by a node contains its adjacent links, every second update contains the links to its neighbors, every third update contains the links to nodes that are two hops away and so on. As a result, a significant fraction of link state entries is suppressed in a typical update.

Because of the update strategy of FSR, the sender may have imprecise knowledge of the best path to the destination but the route is expected to become more accurate as the packet approaches the destination. In high mobility applications, the update frequency is critical because too slow updates potentially result in highly suboptimal routes and too fast updates result in excessive routing traffic.

OLSR uses the concept of multipoint relays to minimize the size of routing packets and the number of rebroadcasting nodes during each route update. Each node selects a set of neighbors that are responsible for rebroadcasting its routing packets. This set of neighbors is called multipoint relay (MPR) set. The MPR set is selected such that all nodes within the two-hop neighborhood must have a bidirectional link to the MPR set. To select the MPR set, each node periodically broadcasts a list of its neighbors. By listening to the neighbor list updates, a node can deduce the set of all nodes that are within two hops and select an MPR set that covers all two-hop neighbors. For an example of an MPR set, see Figure 4.1.

OLSR allows only nodes belonging to some MPR set generate link state up-

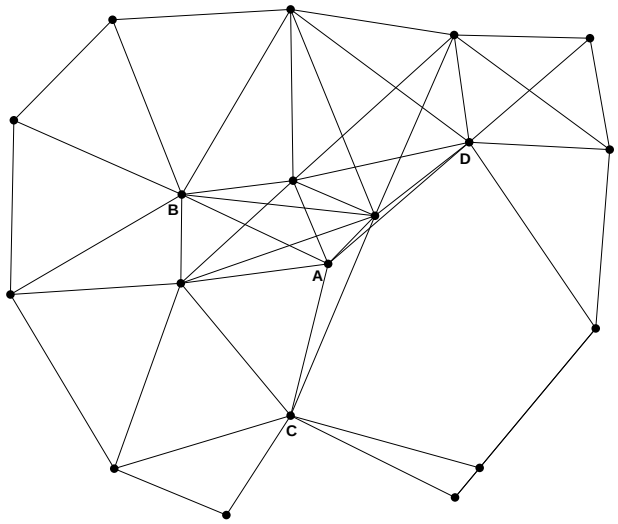


Figure 4.1: Nodes B, C and D belong to the minimum MPR set of Node A.

dates. Because the link state updates contain only the links between MPR nodes and the nodes that have selected them, only partial information is made available at each node. Since it was shown in [49] that there exists at least one path from any node to any other node consisting of only MPR nodes, the partial information is sufficient for local shortest path calculations. OLSR also uses only periodical link state updates with sequence numbering. Thus, the timing of updates is as important as in FSR.

OLSR reduces routing overhead when the network is dense. This can be seen, for example, in Figure 4.1 where Node A has to inform only 3 of its neighbors about link state changes. However, when the network is sparse, a larger fraction of neighbors belongs to the MPR set and the benefit from using OLSR reduces. On the other hand, if the network is very dense, the selection of the MPR set can increase the additional delay.

4.4 Reactive routing

Unlike proactive protocols, reactive protocols do not attempt to maintain a complete knowledge of the network. The routes to destinations are found and maintained only when needed, on-demand. The benefits from on-demand route discovery are the reduction of routing traffic because only some of the nodes need to be informed about topological changes and the decreased storage requirements for route data in nodes. Whenever a node does not have information where to forward traffic packets, it has to start a route discov-

ery procedure. Thus, compared to proactive routing, reactive routing causes additional delay to data packets.

Proactive and reactive routing can be further compared by considering traffic characteristics, such as traffic diversity and session lengths. The traffic diversity means here how uniformly source-destination pairs are distributed in a network. Reactive routing suits well in situations where nodes usually want to communicate with the same, relatively small group of nodes. In these situations, reactive routing considers only changes along the used routes and the overall routing traffic is reduced. On the other hand, when each node has approximately same probability to be selected as a destination, the routing overhead of reactive routing approaches that of proactive routing. When communication session lengths are only a few packets and reactive routing is used, the overhead of the route discovery for each session becomes a burden. During a long session the route found by reactive routing can become suboptimal (e.g., due to node mobility) while proactive routing can adapt to topological changes during the session providing a new optimal route.

Reactive routing protocols can be divided into two classes according to how data packets are handled after the initial route discovery: source routing and hop-by-hop routing. In source routing protocols, each data packet contains a complete source-destination path and the responsibility of intermediate nodes is simply to forward packets according to the path. In hop-by-hop routing, data packets contain only the destination and the next hop addresses. Therefore, intermediate nodes have to decide independently where to forward a data packet. We will present here Dynamic Source Routing (DSR) [50] as an example of a source routing protocol and Ad-hoc On-Demand Distance Vector (AODV) [51] as an example of hop-by-hop routing.

4.4.1 DSR

The idea of DSR is to store multiple alternative routes in a route cache for each destination to avoid the slow route discovery procedure. When a sender wants to transmit a packet to a destination, it first checks if it has already a route entry for the destination in its cache. If the sender does not know any route, it broadcasts a route request packet with a unique id. Each node receiving a route request packet acts as follows:

1. If a route request with the same id has been seen before and the destination address does not match own address, drop the route request packet.

2. Otherwise, if no route to the destination is known, add own address to

the route request and broadcast the route request.

3. If a route to the destination is in the route cache, form a complete route between the sender and the receiver using the route cache entry and the previous hop addresses included in the route request. Send a route reply packet including a complete route to the destination along the chain of previous hops.

4. If the id of the destination matches the local node id, send a route reply packet including the chain of previous hops back to the sender.

Because a destination replies to all received route requests, a source learns many alternative routes that are useful if the shortest route fails. In fact, any node that forwards or overhears any packet adds the source route included in the packet to its route cache. In the case of link failure, the node that notices the broken link sends a route error packet back to the sender and truncates all route cache entries that contain the broken link at that point. All intermediate nodes between the route error sender and the original sender must also truncate their entries accordingly and the sender has to start using an alternative route to the destination. The aggressive route learning and caching of DSR requires a large cache in each node and in order to prevent cache overflow, each route cache entry is deleted after an expiration period.

The route discovery procedure of DSR is based on traditional flooding and it becomes more and more impractical as the number of nodes in a network increases. In large networks, the number of route replies representing alternative routes can also become impractical. This, of course, results in more and more route cache entries in each node. Because of all these reasons, DSR is suitable only for small and sparse ad hoc networks consisting of tens of nodes.

4.4.2 AODV

AODV can be thought of as a reactive version of the DSDV protocol. Each node maintains a monotonically increasing sequence number for itself (increased before sending a route request (RREQ) or a route reply packet (RREP)) and knowledge about local links to its neighbors. The local link knowledge is maintained by a periodic hello packet exchange. Each node includes its id and sequence number and broadcasts the hello packet to all neighbors. In addition, each node maintains a routing table with an entry to all recently needed destinations. The routing table entry contains:

- Destination address,

- Next hop address,
- Number of hops to destination,
- Sequence number for the destination,
- Addresses of the active neighbors, and
- Expiration time for the entry.

A routing table entry is updated only if a control packet has a higher sequence number for the destination or in the case of equal sequence numbers, when the number of hops in the update message is smaller. A neighbor is added to the active neighbor list if it has recently originated or relayed a packet for the destination. The expiration timer for a route entry is reset whenever the entry is needed. Thus, only recently needed routes are stored in the node.

If a source does not know a route to a destination, it initiates the path discovery process by broadcasting an RREQ packet. RREQ includes:

- Source address
- Destination address
- Source sequence number
- Last destination sequence number known to the source
- RREQ id
- Hop count

The intermediate nodes that do not have a route to the destination rebroadcast the RREQ packet, increase the hop count of the RREQ and add or update the routing table entry for the source. To eliminate unnecessary rebroadcasts of RREQ packets, intermediate nodes check the source address and the RREQ id and drop RREQ packets they have seen before.

When RREQ reaches a node that knows a route to the destination (or the destination itself), it checks the destination sequence number from RREQ. If the destination sequence number in the routing table entry is greater or equal than the destination sequence number in the RREQ packet, the node sends an RREP packet to the neighbor from which it received RREQ. RREP packets contain the source and destination addresses, the destination sequence number and the hop count. As the RREP packet travels back to the source, each node along the path increases the hop count and adds or updates the route table

entry for the destination before sending RREP towards the source. The source can add a route table entry for the destination and start transmitting data packets as soon as it receives the RREP packet. In the case of a link failure, the node that detects a broken link sends a RREP packet with an infinite hop count to all active neighbors. Those nodes subsequently relay that packet to their active neighbors and so on.

AODV has less overhead than DSR because routing and data packets do not include the complete route information. However, AODV has the same scalability problems as DSR because the route discovery is based on flooding. However, the scalability of AODV and other routing protocols that use flooding can be improved by utilizing the advanced broadcasting techniques described in Section 4.2 and in [45].

4.4.3 Hybrid protocols

As discussed earlier, reactive routing protocols generate less routing traffic overhead than proactive protocols with the cost of increased delay caused by the on-demand route discovery. The idea of hybrid routing protocols is to combine proactive and reactive routing in such a way that the performance, measured, for example in throughput and delay, is better than in pure proactive or reactive protocols. We present here two hybrid routing protocols, Zone Routing Protocol (ZRP) [52] and AntHocNet [53], which combine proactive and reactive routing in very different ways.

In ZRP, each node has a routing zone that includes all nodes that are within the distance of the zone radius, r_{zone}. The zone radius is usually given in number of hops. The nodes that are exactly r_{zone} hops away from a source or destination are called border nodes. The idea of ZRP is that each node maintains complete topological information within its routing zone using proactive routing. Thus, a source has a route ready in its routing table when it wants to communicate with a destination within its routing zone. Figure 4.2 illustrates a part of a network with $r_{\text{zone}} = 2$.

When the destination is not within the routing zone of the source, the source sends a RREQ packet to all its border nodes. Now, the border nodes check whether the destination is within their routing zone. If a border node sees RREQ for the first time and it has no routing table entry for the destination, it adds its own address into the RREQ packet and resends RREQ to all its border nodes (excluding the node from which it received RREQ). When a border node has the destination within its routing zone, it adds the path of border nodes from the source to itself into a RREP packet and sends it back to the source using the same path. For example in Figure 4.2, data packets

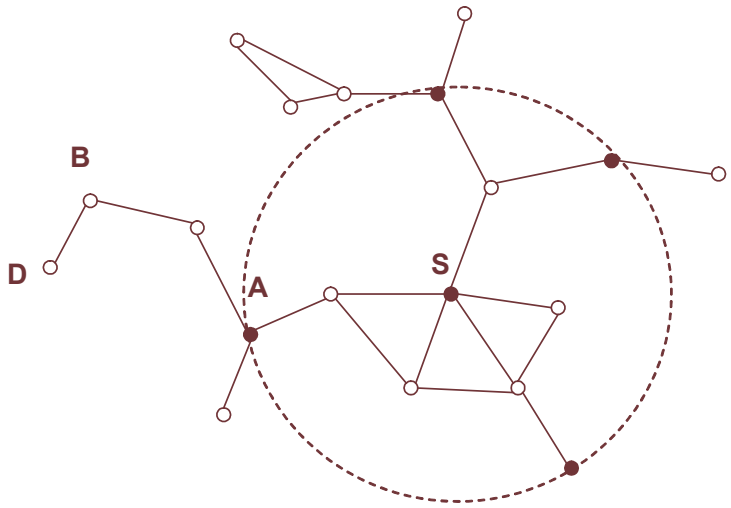


Figure 4.2: A part of a network with $r_{\text{zone}} = 2$. The border nodes of Node S are drawn in black.

from Node S to Node D are routed along the path S-A-B-D.

The choice of the zone radius has a great impact on the operation of ZRP. If $r_{\text{zone}} = 1$, ZRP works as a pure reactive protocol and correspondingly if $r_{\text{zone}} = \infty$, ZRP works as a pure proactive protocol. The increase in r_{zone} decreases delay but adds routing overhead. Correspondingly, a smaller zone radius would reduce routing overhead and increase delay. Ideally, the zone radius could be adjusted depending on the application and the traffic distribution. However, this can be very hard to do in a dynamic way.

AntHocNet is a new routing protocol based on the framework of Ant Colony Optimization (ACO). ACO algorithms are inspired by the behaviour of natural ant colonies during food searching process. While walking between a food source and the nest, ants spread a chemical called pheromone on the paths they take. At path crossings, ants choose a path stochastically preferring paths with a higher amount of pheromone. In addition to routing in data networks, ACO algorithms are also applied in solving NP-hard problems, such as the traveling salesman and the quadratic assignment problems [54].

Routing tables in AntHocNet are called pheromone tables. A pheromone table entry has a destination address, a next hop towards the destination and a pheromone value for the route. The pheromone value that indicates the goodness of the route is inversely proportional to the expected time it takes to reach the destination through the next hop. There can be multiple routes to a destination in a pheromone table and in that case, the routing decision is done stochastically. The probability that a given route is chosen is weighted

with the pheromone value of the route.

AntHocNet has a route discovery procedure somewhat similar to the reactive protocols described earlier. A source broadcasts a route request packet called a reactive forward ant (RFA) in order to find the destination. A node receiving an RFA for the first time either forwards it according to its pheromone table towards the destination or if it has no entry for the destination, rebroadcasts it. Copies of the same RFA arriving later at the node are discarded and similarly the destination only processes the first RFA. Each RFA stores the complete traveled route. Thus, when the first RFA reaches the destination, it has a backward route ready to the source. As a response, the destination sends a reactive backward ant (RBA) back to the source. As the RBA travels the route to the source, it collects information on how long it has to wait at each node. At the arrival of an RBA, intermediate nodes can update their pheromone tables according to this information. When RBA arrives at the source, the source can add a pheromone table entry for the destination and start transmitting data packets.

In AntHocNet, the route maintenance and the discovery of new routes are done proactively with the help of proactive ants and periodic route advertisements exchanged between neighbors. The periodic route advertisements include only the destinations to which data packets have been recently forwarded. Only the route with the best pheromone value is advertised. The interval between route advertisements is kept long to avoid excessive routing overhead. Thus, the pheromone information from advertisements can be outdated and is not directly used in data packet routing. The information from periodic updates is stored in a virtual pheromone table that is used to route only ants but not data packets. If there is a significantly better route in the virtual pheromone table than in the used pheromone table, a proactive ant is sent towards the destination. As the ant returns, the node can either add a new route to its pheromone table or update existing routes.

Because the pheromone values for congested hops decrease, AntHocNet is able to spread data load dynamically, thus reacting to congestion. In [53], AntHocNet was compared to AODV in three scenarios with different node densities and mobility patterns. It was observed that the average delay and the ratio of correctly sent packets was better in AntHocNet. There was more routing overhead in AntHocNet with a small number of nodes, but as the number of nodes was increased, the routing overhead in AODV exceeded the overhead in AntHocNet.

4.5 Geographic routing

Routing protocols presented in Sections 4.3 and 4.4 collect topology information from the network and store it in routing tables. The amount of control traffic needed to update routing tables increases rapidly when the number of nodes in the network is increased. It is questionable if any of the previously presented protocols are able to handle the routing task in a network with thousands of mobile nodes.

A completely different approach to routing in ad hoc networks is to make a forwarding decision based on the geographical locations of nodes. As long as the transmitting node knows its own and the destination's location in addition to the locations of its neighbors, the forwarding decision can be done completely locally without any topological information stored in a routing table. Geographic routing assumes that each node can find out its own location with the help of the global positioning system (GPS) or some other localization technique. When this location information is exchanged with neighbors, each node learns the locations of its neighbors. The constraint on the scalability of geographic routing is how the location of a destination is made available at a source. For this purpose, numerous location service protocols have recently been proposed.

Most geographic routing protocols simply forward packets to a neighbor that is closest to the destination. This kind of routing method and its variants are commonly called greedy forwarding. Greedy forwarding methods perform well in dense networks but fail to reach the destination if packets are forwarded to a concave node that has no neighbors nearer to the destination than itself. Thus, if a geographical routing protocol is guaranteed to deliver packets to all connected destinations, it has to include a method to route around concave nodes.

4.5.1 Greedy forwarding methods

Greedy forwarding methods forward packets to a neighbor that is closest to the destination measured in progress, Euclidian distance or direction. Progress is defined as the distance between Node S and the projection of its neighbor A onto to the line connecting S and the destination D. Most Forward within Radius (MFR) [55] was the first proposed geographic routing method for ad hoc networks. MFR simply forwards packets to the neighbor with the maximum progress with respect to the sender and the destination. If there are no neighbors with a positive progress, the neighbor with the least negative progress can also be chosen. This can be beneficial in some situations like

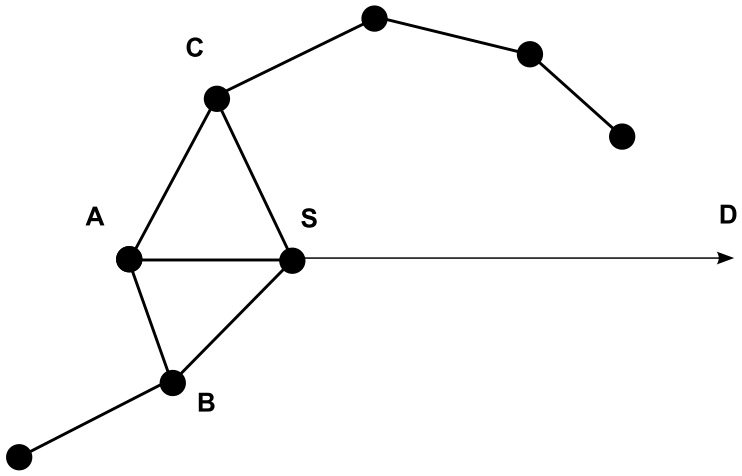


Figure 4.3: The situation when it is beneficial to allow a concave node (S) forward packets backwards. Packets from A and B would be stuck in S unless S forwards them to its neighbor with the least negative progress, C.

the one depicted in Figure 4.3, but usually the result is a local routing loop between a concave node and its backward neighbor.

Another obvious routing metric to be used in geographic routing is the Euclidian distance. Geographical Distance Routing (GEDIR) [56] forwards packets to the neighbor that is closest to the destination regardless of whether it is closer than the sender or not. Thus, a packet can also be forwarded in the backward direction. This can potentially lead to a routing loop between two neighboring nodes. Similar to MFR, routing backwards can be useful in some situations (Figure 4.3) and in harmful situations, two node loops are easy to detect. Although the next hop selection criterion is very similar in MFR and GEDIR, the selected neighbor in MFR may also be farther from the destination than the next hop in GEDIR (Figure 4.4).

The third approach to greedy forwarding is to forward to the neighbor that is closest to the line drawn between the source and the destination. This approach is applied in the compass routing method [57] where the sender S selects the next hop A to the destination D such that the angle $\angle ASD$ is minimized. For example, let us consider the situation in Figure 4.4. Compass routing selects Node C as the next hop because $\angle CSD < \angle BSD < \angle ASD$. It was shown in [56] that if compass routing is allowed to forward backwards (to nodes with $\angle ASD > 90°$), loops between more than two nodes can occur. For an example of this, see Figure 4.5.

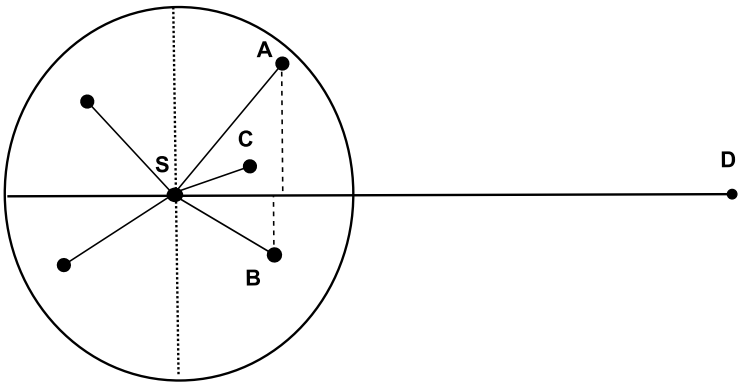


Figure 4.4: The choice of a next hop from Node S towards the destination D in different greedy forwarding methods. S selects A in MFR, B in GEDIR and C in compass routing.

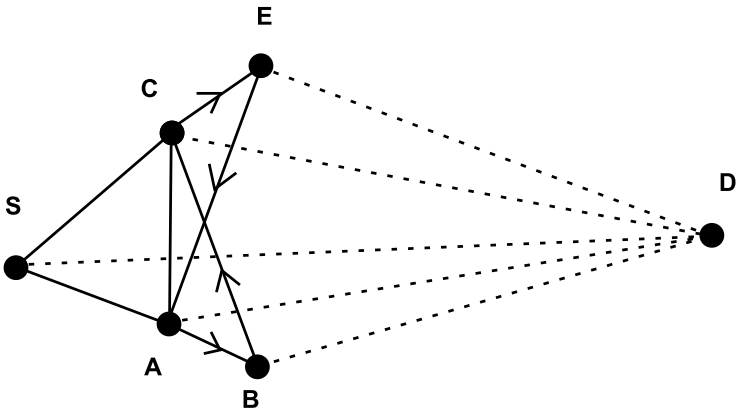


Figure 4.5: A four node loop in compass routing. A packet sent from S towards D follows the path S-A-B-C-E-A-B-C-E-...

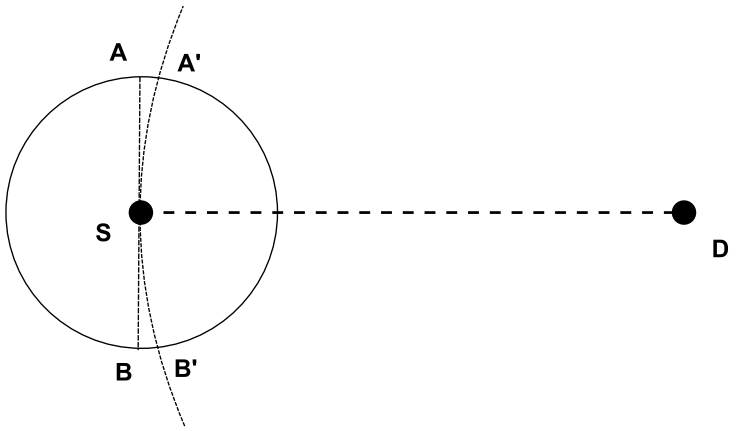


Figure 4.6: Forward and backwards areas for different greedy forwarding methods. In MFR and in the compass routing, S is a concave node with respect to D if it has no neighbors in front of the line AB. In GEDIR, S is a concave node with respect to D if it has no neighbors in front of the arc A'B'.

4.5.2 Routing around concave nodes

A concave node is defined as a node that has no forward neighbors. Depending on the routing strategy, the forward and backward areas are slightly different. When MFR is used, a node is concave if it has no neighbors with a positive progress. When compass routing is used, a node is concave if all angles between the sender, any neighbor and the destination are greater than 90°. For both MFR and compass routing, the border between forward and backward areas is a line perpendicular to the line between the sender and the destination dividing the transmission area around the sender into two half-circles. When GEDIR is used, a node is concave if it has no neighbors nearer to the destination than itself. The border between forward and backward areas is now an arc of a circle centered at the destination with a radius equal to the distance between the sender and the destination. See figure 4.6 for an example.

The number of concave nodes with respect to a given destination is a function of network density. In sparse networks, the percentage of concave nodes is high and as a result greedy forwarding methods often fail to reach a destination. According to the simulations in [56], the delivery rate, defined as the ratio of numbers of packets received by the destination and sent by sources, was about 90 % for an average degree (number of neighbors) of 8 but only 50 % for an average degree of 4 for all greedy forwarding methods. To avoid routing failures in sparse networks, a number of routing algorithms have been proposed that either avoid forwarding to concave nodes or route around concave nodes and continue using greedy forwarding when possible.

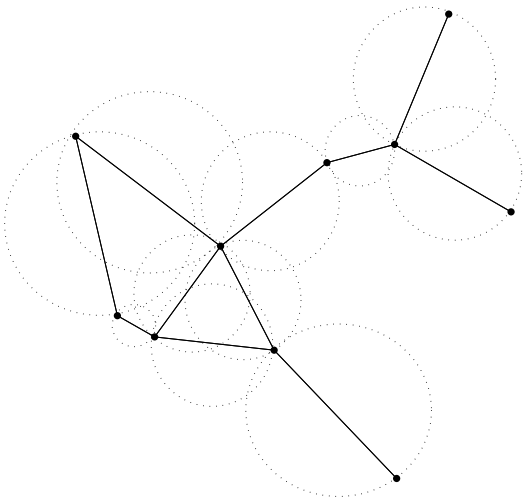


Figure 4.7: A Gabriel graph. There are no nodes located in any disk with a line between two Gabriel neighbors as its diameter.

Greedy/flooding [56] is a simple method to deal with concave nodes. When a node receives a packet and finds out that it has no forward neighbors with respect to the destination, it broadcasts a packet declaring its concavity to all its neighbors and rejects further copies of the packet with the same id. Neighbors then add the concave node into a temporary list of concave neighbors with respect to the given packet. Nodes in the concave neighbor list are ignored in forwarding decisions. Because concavity information is advertised independently for each packet, the greedy/flooding method is suitable for sparse networks with frequent topological changes and short packet flows. When the network topology is stable and packet flows between a source and a destination are long, the greedy/flooding method results in excess routing overhead because the same concavity information is advertised for each packet belonging to the same flow.

Face routing [58] is a method to route around concave nodes without any need for memory in nodes. The idea of face routing is to find a planar subgraph of the network graph and apply simple routing algorithms on this subgraph. A planar graph is defined as a graph in a plane without any edge crossings. Face routing uses a Gabriel graph [59] as a planar subgraph. The Gabriel graph is defined as follows: given any adjacent nodes A and B in a graph, the edge AB belongs to the Gabriel subgraph if no other nodes are located in the disk with the line AB as its diameter (see Figure 4.7).

In face routing, each node calculates its Gabriel neighbors locally before making a routing decision. Node i calculates its Gabriel neighbors using the following simple algorithm:

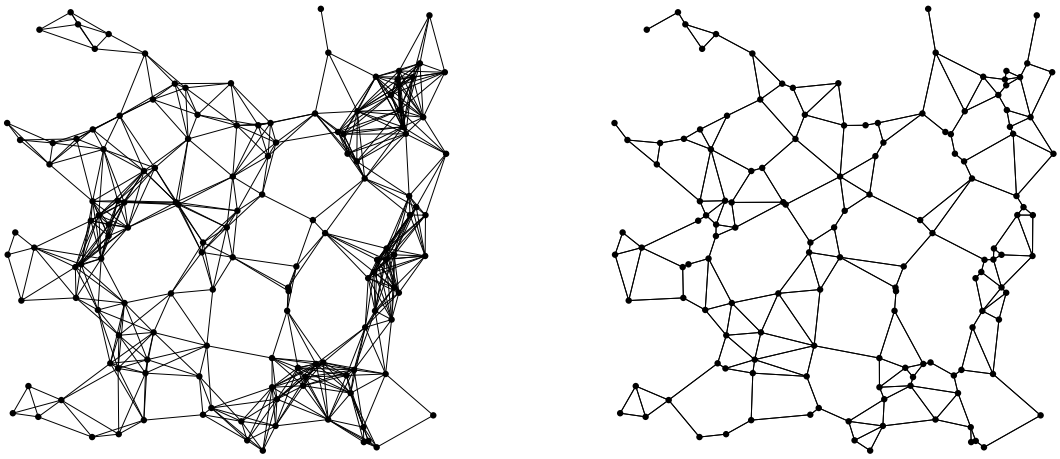


Figure 4.8: A network graph and the corresponding Gabriel graph.

1. For each neighbor j, calculate $l(i,j)^2$.
2. If for each other neighbor $k \neq j$, $l(i,k)^2 + l(j,k)^2 \geq l(i,j)^2$, nodes i and j are Gabriel neighbors.

As it can be seen from Figure 4.8, there are no intersecting edges and the overall number of edges is remarkably reduced. A Gabriel graph partitions the plane into faces that are bounded by polygons made up of graph edges.

The face routing algorithm routes a packet along interiors of the faces intersecting the line between the source and the destination. Faces are traversed using the well-known right hand rule: a packet is forwarded along the next edge clockwise from the line that is drawn normal to the edge of arrival. The operation of the face routing algorithm from source s to destination d is as follows:

1. Set $p = s$.
2. While $p \neq d$,
   (a) Traverse the face until reaching edge (u, v) that intersects line $\overline{pd}$ at some point $p' \neq p$.
   (b) Set $p = p'$.

The face routing algorithm is most easily understood from the packet's point of view. Let us consider, for example, the route from S to D in Figure 4.9. The packet keeps its "right hand" on the wall (edge) and ignores edges that intersect with the line between S and D.

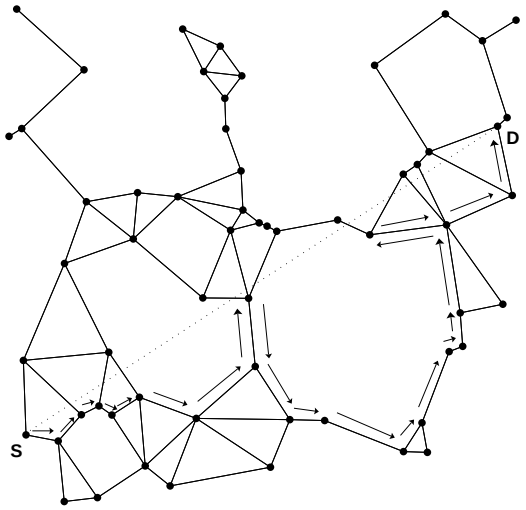


Figure 4.9: The route from S to D using face routing with the right hand rule.

Although face routing is proved to be loop-free and guaranteed to deliver a packet in a connected network, sometimes routes can be extremely long. For example, consider the route between S and D in Figure 4.9 if the left hand rule was used. To avoid long routes, Bose et al. proposed Greedy-Face-Greedy (GFG) routing algorithm [58]. In GFG, GEDIR forwarding is used (using the normal network graph) until the packet reaches a concave node. Then the face routing algorithm is applied until the packet reaches a node that is closer to the destination than the concave node. Thus, GFG uses face routing only as a method to route around concave nodes and returns to greedy forwarding as soon as possible.

In GFG, the efficiency of face routing is unpredictable because the number of hops in a face route can be high. Greedy Other Adaptive Face Routing (GOAFR) [61] was proposed to restrict the face traversal inside an adaptively changed ellipse region. If the border of the ellipse is hit during the face traversal, the traversal is performed in the opposite direction. If the border of the ellipse is hit again, the size of the ellipse is increased and the face traversal algorithm is performed again.

We will present here the improved version, GOAFR+ [62] that uses a circle instead of an ellipse to restrict the face traversal. GOAFR+ requires setting constant parameters ρ_0, ρ and σ before the algorithm is started. In order to GOAFR+ to work correctly, the parameters have to satisfy $1 \leq \rho_0 < \rho$ and $\sigma > 0$. The operation of GOAFR+ from source s to destination d is as follows:

1. Set $p = s$. Initialize circle C centered at d with radius $r_C = \rho_0 l(s, d)$.

2. Repeat greedy forwarding until reaching either d or concave node i. During greedy forwarding, reduce the radius of C ($r_C = r_C/\rho$) whenever the currently visited node is within C. If d is reached, the algorithm ends. Otherwise, go to step 3.

3. Traverse face F_i containing line $\overline{id}$ completely. During the face traversal keep track of the number of nodes closer to d than i, denoted by p, and the number of nodes not closer to d than i, denoted by q. When face F_i is completely traversed, advance to the node belonging to F_i that is closest to d and go to step 2. There are four special cases when the whole face is not traversed:

   (a) If $p > \sigma q$, advance to the node belonging to F_i that is closest to d. Go to step 2.

   (b) When the next traversed node would be outside C and C is encountered for the first time in F_i, turn back and traverse F_i in the opposite direction.

   (c) If C is hit for the second time, advance to the node belonging to F_i that is closest to d. Go to step 2.

   (d) If C is hit for the second time and no visited node is closer to d than i, increase C ($r_C = \rho r_C$) and go to step 3.

Step 2 in the GOAFR+ algorithm reduces the size of the circle restricting the face traversal during greedy forwarding as a packet approaches the destination. If the circle is too small and face routing cannot find a node closer to d than i, the size of the circle needs to be increased. This is done in step 3.(d). As can be seen from step 3.(a), the choice of σ determines how soon the algorithm switches from face routing back to greedy forwarding. If σ is small enough, greedy forwarding is proceeded as soon as the packet enters the first node that is closer to d than i.

The operation of GOAFR+ using parameters $\rho_0 = 1.4$, $\rho = \sqrt{2}$ and $\sigma = 1/100$ is illustrated in Figures 4.10 and 4.11. Figure 4.10 depicts greedy forwarding phases between source S and destination D. Note that the circle restricting the face traversal decreases as the packet approaches D. In the middle of the route, greedy forwarding is not possible because of concave node A. At node A, GOAFR+ switches to face routing that is depicted in Figure 4.11. The packet is routed according to the right hand rule until it meets an edge intersecting with the circle border. At that point, the face traversal direction is reversed and the packet traverses along the face until it reaches node B that is closer to D than A. The rest of the route is traversed using greedy forwarding.

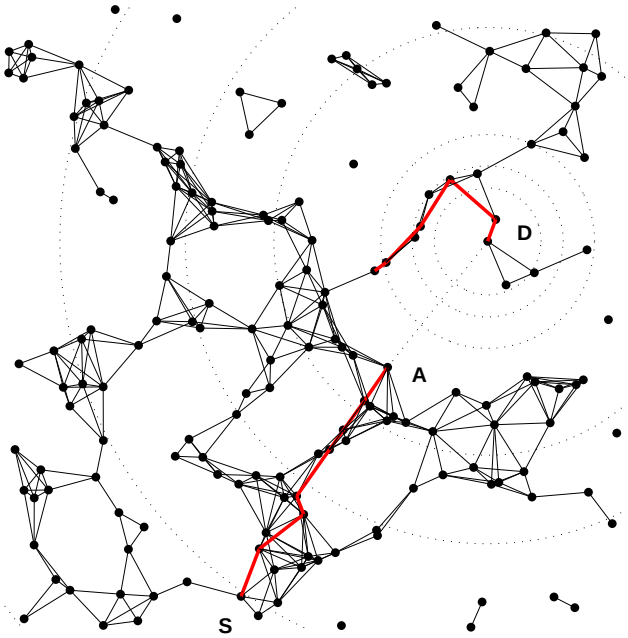


Figure 4.10: The greedy forwarding phases of GOAFR+ between source S and destination D. The greedy forwarding path is depicted in red and the circles restricting the face traversal are dotted.

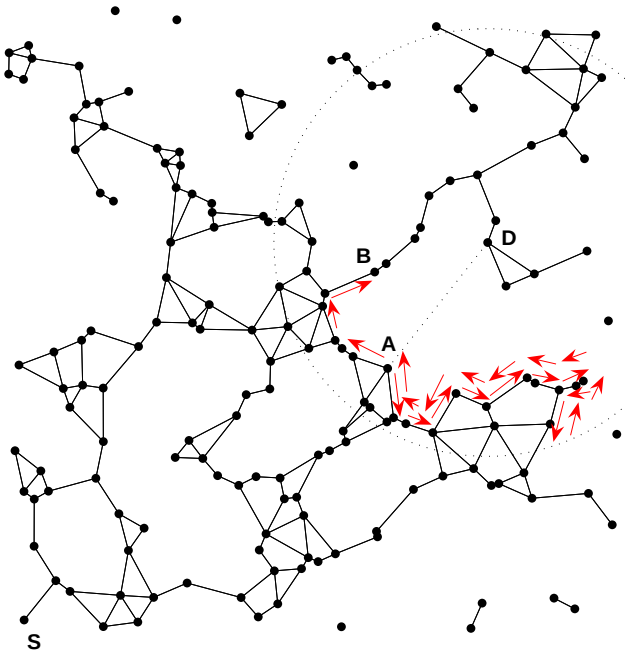


Figure 4.11: The face routing phase of GOAFR+. The face traversal path is plotted with red arrows.

In order to measure the average case efficiency, GOAFR+ was simulated in a static network. With simulation parameters $\rho_0 = 1.4$, $\rho = \sqrt{2}$ and $\sigma = 1/100$ and the path length as a performance metric, GOAFR+ outperformed GFG in sparse networks. When the network density is increased, both algorithms approach greedy forwarding (GEDIR) and there is no difference in performance.

4.5.3 Greedy forwarding with a jointly designed MAC scheme

The greedy forwarding methods presented in Section 4.5.1 always try to forward packets to the best neighbor. If the best neighbor also hears other transmissions, a collision occurs and no progress is made. In those cases, it would have been more beneficial to forward the packet to a forward neighbor that is able to receive it. We will now present two protocols that utilize the previous idea by combining greedy forwarding with a jointly designed MAC scheme.

The idea of Extremely Opportunistic Routing (ExOR) [63] is to broadcast a packet to all forward neighbors and use a slotted acknowledgement scheme to find out the best candidate. Before broadcasting a packet, a sender adds a forwarding candidate list into it. The forward candidate list includes all forward neighbors sorted such that the neighbor closest to the destination gets the highest priority. In a later publication [64], the authors suggest using an empirical average delivery rate as a priority metric instead.

After transmission, each forward neighbor that is not hearing another transmission receives the packet. Instead of immediate transmission of ACKs, there is a pre-defined delay related to the forward candidate list entries such that the highest priority candidate gets to send its ACK first. Each node listens to ACK traffic and adds the id of the highest priority successful recipient known to it to the ACK packet. When the total ACK exchange period announced in the data packet has expired, nodes that have not received ACKs containing the id of a higher priority candidate forward the packet further.

Including the id of the successful recipient with the known highest priority into ACK packets helps to reduce duplicate forwarding. This is illustrated in figure 4.12. Node S sends a data packet to nodes A, B and C to be forwarded towards destination D. Node A sends its ACK first then node B and finally node C. Since nodes A and B are not able to hear each other's ACKs, node B would have also forwarded the data packet, if node C did not have included A's id in its ACK. However, duplicate packets are still possible. If nodes A and B are the only nodes hearing a data packet, they both end up forwarding the data packet further.

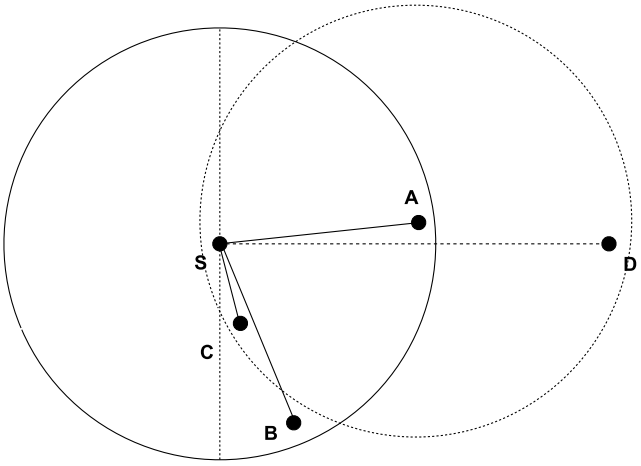


Figure 4.12: The forwarding candidate set {A,B,C} of node S.

The drawbacks of ExOR are the occasional duplicate packets and the need for accurate synchronization within forward candidate sets. Because of the need for accurate synchronizatíon, ExOR is most easily implemented in systems that use GPS for receiving location and accurate timing information.

Beacon-less routing (BLR) [65] is another greedy forwarding scheme that modifies the underlying MAC protocol in order to improve throughput. BLR selects a forwarding node among neighboring nodes without having information about neighbors' location or even existence. Thus, there is no need for periodical location information exchange between neighbors. This is especially beneficial in applications requiring long battery lifetimes.

Before the operation of BLR can be presented, a concept of forwarding area has to be defined. A forwarding area of a node is a region towards a destination such that all nodes within it can hear each other. In addition, a forwarding area should be as large as possible in order to increase the probability of finding a node within the area. Figure 4.13 presents two possible forwarding areas. According to the simulations in [65], a circle is preferred because it has the largest possible area still fulfilling the definition for a forwarding area and there is no significant reduction in the average progress compared to other area choices.

When a node has a packet to send towards a destination, it adds its own coordinates to the packet before broadcasting it to all its neighbors. All nodes receiving a packet can find out using the locations of the previous node and the destination whether they are within the forwarding area. Nodes outside the forwarding area drop the received packet. Nodes inside the forwarding area calculate a forwarding delay that is inversely proportional to the progress from

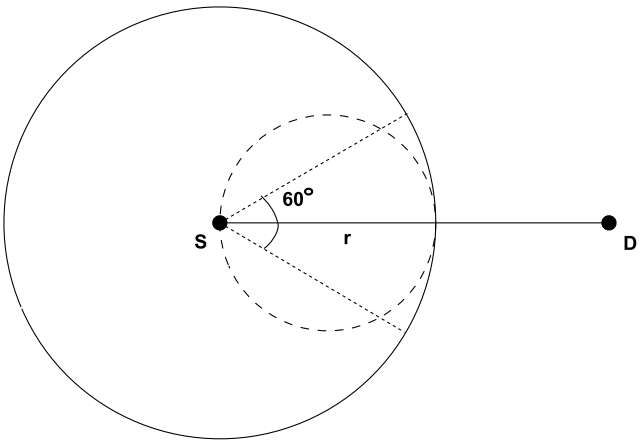


Figure 4.13: Two forwarding areas from S towards D. The 60° degree sector is depicted with dotted lines and the circle with diameter r is dashed.

the sender. A most forward node has the smallest delay and gets to forward the packet first. Other nodes within the forwarding area hear the transmission and cancel their scheduled transmission of the same packet. Note that also the sender hears the transmission and no additional ACK is needed (except from the final destination to the previous node).

As ExOR, BLR requires accurate synchronization for efficient operation. The advantage of BLR compared to other previously presented greedy forwarding methods is that it does not require the exchange of location information in the neighborhood. This is achieved at the cost of smaller forward regions and more frequent greedy forwarding failures. Thus, BLR is most suitable for dense networks. In sparse networks, BLR has to rely often on face routing as a recovery method.

4.5.4 Location service protocols

If each source had the location of each destination available always when needed, geographic routing would be easy, fast and efficient in terms of the path length and scalability. However, the delivery of the location information from a destination to a source is not an easy task and it can result in considerable amounts of delay and overhead traffic. The task of locating the destination is accomplished by a location service that is either a distinct protocol or integrated into the routing protocol.

It is usually assumed that each node finds out its own location using the GPS. However, GPS receivers can be too expensive for some applications, such as large-scale sensor networking. For these applications, there are multiple lo-

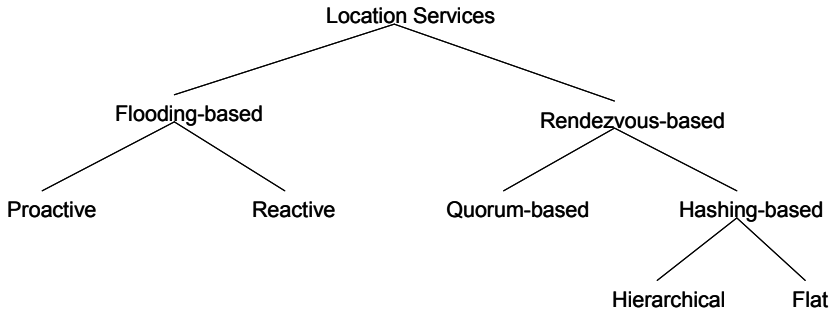


Figure 4.14: A taxonomy of location services [67].

calization algorithms that enable nodes to determine their relative locations automatically. For a survey of localization algorithms, see [66]. When each node knows its own location, the location information is periodically broadcasted to all neighbors. Periodical location updates allow each node to learn the locations of its neighbors that are needed to make greedy forwarding decisions.

Figure 4.14 depicts the taxonomy of location services proposed in [67]. Location services are classified into two general approaches: flooding-based protocols and location database protocols. In flooding-based protocols, each node floods its location to other nodes either periodically (proactive) or when queried (reactive). An example of proactive flooding-based protocol is DREAM [68]. DREAM tries to reduce the overhead of location updates by sending location updates to distant nodes less frequently than to closer nodes and by adapting the location update rate with respect to the node mobility. It is clear that the overhead of flooding-based location service protocols grows very fast as the number of nodes in the network is increased.

In location database protocols, each node acts as a location database server for a set of nodes. When a node moves to a new location, it updates its location to its location update servers. Correspondingly, when a node needs to know the location of a destination, it sends a query to its location query servers hopefully getting an up-to-date response from at least one of the location servers. In order to be sure that the location of any node is known by at least one location server, all nodes have to agree upon a mapping that maps each node's id to one or more other nodes. The way that the mapping is done distinguishes quorum-based protocols from hashing-based protocols.

In quorum-based protocols, location updates from a node are sent to a subset of all nodes and location queries for the node are sent to a different subset of nodes. In order to a query to be successful, the intersection of these two subsets has to be non-empty. As an example of a quorum-based protocol we

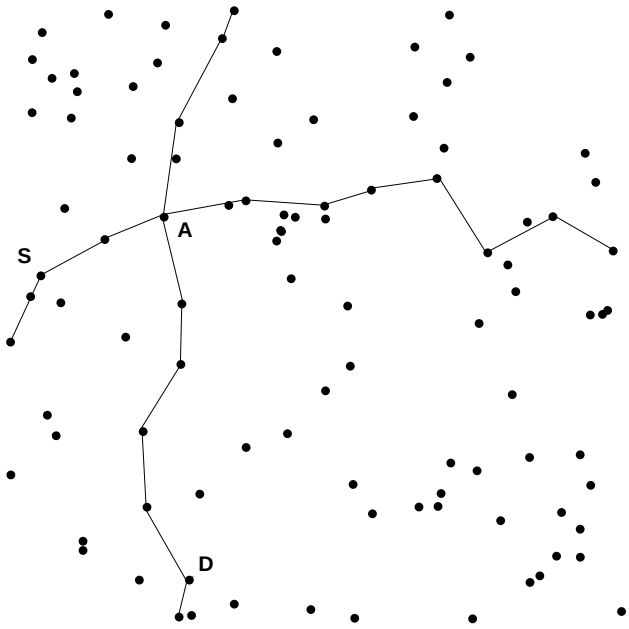


Figure 4.15: A location update column for destination D and a query row for source S. The location update column and the query row intersect at node A.

will shortly present the row-column location service scheme [69]. The idea of the row-column scheme is that a node forwards its location updates to all nodes located in a north-south column with a certain thickness. Correspondingly, a node sends location queries to all nodes located in a west-east row with a certain thickness. Because each query row intersects with each update column (without forwarding failures), location information of all nodes is available for each node.

Update packets are routed using normal greedy forwarding and for query packets, algorithms based on face routing can also be used. Thus, the thinnest possible column is only a single path from north to south. This is depicted in Figure 4.15. A single path may not be able to reach the borders of the network because of concave nodes. On the other hand, when a thicker column is used, the update overhead is increased. Each node keeps a location update table including the time of arrival for the location updates it receives. A query packet contains the latest known location for a destination. If a node in a query row finds out that it has more recent location information than the query packet, it sends a reply packet back to the source.

In hashing-based protocols, the id of each node is mapped to a certain area or to a set of other nodes' ids using a hashing function. The hashing function

is used to choose location servers for a given node id. In order to distribute queries evenly, the hashing function should guarantee that on the average each node acts as a location server for the same number of nodes. Hashing protocols are further divided into flat and hierarchical depending on whether location servers form some kind of hierarchy or not.

A good example of a flat hashing protocol is Scalable Location Update-Based Routing Protocol (SLURP) [70]. The hashing function of SLURP maps node ids to a geographic sub-region of the network. Any node in a sub-region is responsible for storing locations for all nodes that are mapped to the sub-region. In practice, this means that all location servers for a node are located within the same sub-region called the home region. Because location updates, queries and data packets are geographically routed towards sub-regions (instead of exact node locations), each node must have a table that contains static coordinates for all sub-regions.

A node sends a location update only when it moves from a sub-region to another. The location update packet is broadcast to all nodes in the new sub-region and geographically routed towards the home region. When a node belonging to the home region receives the location update packet, it further broadcasts the update to all nodes within the home region. When a node needs to know the location of a destination, it first calculates the home region of the destination using the hash function and destination's id. A query packet is then sent using geographic routing towards the home region of the destination. The first node within the home region of the destination that receives the query replies with the destination's location. When the source node receives the reply, it is ready to start data transmission. A data packet is geographically forwarded until it is received by a node within the sub-region of the destination. Once the packet is inside the correct sub-region, SLURP proposes that source routing (similar to DSR) is used to reach the destination.

The critical design parameter in SLURP is the size of sub-regions. Small sub-regions are preferred because source routing becomes inefficient if the number of nodes within the sub-region is high. On the other hand, large sub-regions would reduce the location update traffic and the probability that a sub-region is empty would be small. The disadvantage of SLURP is that the location of a node is independent of its home region location. Even if the hashing function is cleverly selected so that the initial distance from nodes to their home region is small, nodes can move far from their static home regions during their lifetime.

Most of recently proposed location service protocols belong to the group of hierarchical hashing-based protocols. One of the early proposals is Grid Location Service (GLS) [71] whose idea has been later improved, for example, in [72] and in [73]. GLS assumes that all nodes know the same global parti-

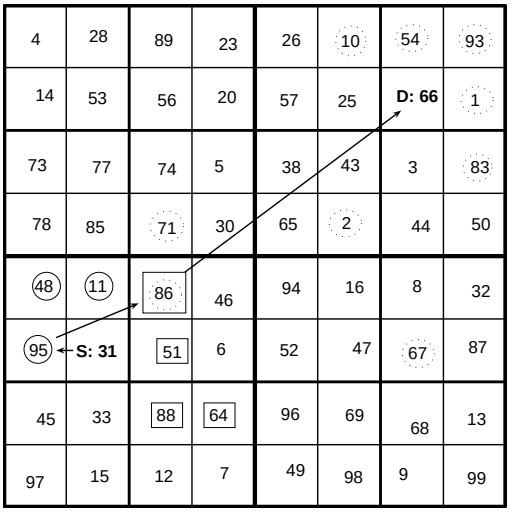


Figure 4.16: A location query process from S to D. D's location servers are marked with dashed circles, the sibling order-1 nodes to which S acts as a location server are marked with full circles and the sibling order-2 nodes to which node 95 acts as a location server are marked with rectangles. For simplicity it is assumed that there is only one node within each order-1 square.

tioning of the world into a hierarchy of grids with squares of increasing size. The smallest square is called an order-1 square, each order-2 square consists of four non-overlapping order-1 squares, each order-3 square consists of four non-overlapping order-2 squares and so on. The four order-$(n-1)$ squares sharing the same n square as the parent square are called sibling squares. Each node has a location server in each sibling at each hierarchy level. For example in Figure 4.16, node D has 3 order-1 location servers {1,54,93}, 3 order-2 location servers {2,10,83} and 3 order-3 location servers {67,71,86}. GLS assumes that each node knows the locations of all nodes within an order-1 square. This can be achieved by periodic broadcasting inside order-1 squares. All other packets in GLS are routed using geographic routing.

GLS employs simple hashing when selecting location servers. Within each sibling square node A chooses the node whose id is next from its own id in the circular id space. For example, if there was a set of possible location servers with ids {3, 7, 15, 37, 58, 82}, node 24 would select node 37 as its location server and node 90 would select node 3 as its location server.

Let us consider the case where each node knows the locations of those nodes that have chosen it as their location server. When source S needs to know the location of destination D, it first sends a query to node A with the next id from D within its order-1 square. Node A then sends the query to node B

whose location A knows and that has the next id from D within A's sibling order-1 squares. In the same way, node B resends the query to node C whose location it knows and that has the next id from D within B's sibling order-2 squares. This is repeated until the query is received by one of D's location servers. For example in Figure 4.16, because S is the only node within its order-1 square, it sends the query to node 95 (next id from 66) within its sibling order-1 squares. Node 95 then sends the query to node 86 within its sibling oder-2 squares. Node 86 happens to be D's location server and so node 86 can send the query directly to D. S can start data transmission as soon as it receives a reply from D.

Because nodes do not know the locations of their location servers, location updates must be routed in the same way as query packets. In addition to the id of a location server, location update packets include the location of the update source. To avoid unnecessary location update traffic, a node updates its location servers in order-i sibling squares only after it has moved a distance of $2^{i-1}d$, where d is the update threshold.

It was proven in [71] that a query needs no more than n location query steps (transitions between sibling squares) to reach its destination when both the source and the destination are within the same order-n square. In addition, it holds that the query never leaves the order-n square from which it starts. However, these theorems only hold for static networks and in dynamic networks, queries can result in failures due to node mobility, sleep periods and node failures.

4.6 Routing in sensor networks

As already mentioned in Section 2.4, there are differences between ad hoc and wireless sensor networks that require special attention when designing techniques for wireless sensor networking. The most important differences are the following:

- the number of nodes in sensor networks is usually larger demanding better scalability from routing protocols,
- energy conservation and network lifetime maximization are bigger issues, and
- topology changes are usually caused by the on-off periods of nodes instead of node mobility.

In addition to these three differences between ad hoc and sensor networks, there are other characteristics in sensor network communications that are directly related to routing.

In sensor networks, communication is usually data-centric instead of node-centric. The data-centric communication means that it is more important to receive certain data than to know which of the nodes sent the data. The traffic pattern in sensor networks is typically many-to-one as a set of sensor nodes transmit sensed data towards a single sink node. Finally, usually the same phenomenon is sensed by multiple sensor nodes and thus the collected data has some redundancy. Instead of sending all the redundant data, it conserves energy and bandwidth to aggregate data at intermediate nodes between the examined phenomenon and the data sink.

There exists a wide range of network layer protocols related to routing and topology control. In [74], routing protocols for sensor networks are classified as data-centric, hierarchical and location-based protocols. In data-centric protocols, there are two basic approaches on how the sensed data is disseminated. Either the sink broadcasts a query for certain data and nodes having the data transmit towards the sink, or sensor nodes advertise for the availability of certain data and interested nodes send requests for the data. Hierarchical protocols form clusters of sensor nodes and select a cluster head for each cluster. A cluster head aggregates data from its cluster and sends the aggregated data to the sink either directly or via other cluster heads. Location-based routing protocols for sensor networks are very similar to those designed for ad hoc networks presented in Section 4.5. However, in addition to having a path length or the number of hops as a routing metric, they aim to minimize the consumed energy on a route or to maximize network lifetime.

Due to size limitations in this study, we consider here only the routing protocols designed for large-scale sensor networks (the distance between the sink and the examined phenomena can be great) that consists of large number of identical nodes. Due to this limitation, hierarchical protocols are not considered because most of them either assume that each sensor node is able to communicate with the sink directly [75, 76] or assume that there are special nodes with less energy constraints acting as gateways or routers [77]. More precisely, we present two examples of data-centric routing methods: directed diffusion [78] and Sensor Protocols for Information via Negotiation (SPIN) [79]. Finally, we present a topology control algorithm [80] to be used together with location-based routing, which compute an energy-efficient subnetwork, the minimum energy communication network, for a sensor network.

4.6.1 Data-centric routing

Directed diffusion is a data-centric routing method consisting of three phases: interest broadcasting, data forwarding and path reinforcement. In directed diffusion, data is named using attribute-value pairs describing, e.g., the type of data, the sending rate of data and the area from which data is collected.

When a sink needs certain data from the network, it broadcasts an interest with certain attribute-value pairs. A node receiving the interest checks its interest cache whether it already has an entry for the interest. An entry in an interest cache includes attribute-value pairs defining the interest and several gradient fields. A gradient field includes the id and the requested data rate of the node from which the interest was received and a lifetime for the gradient. If the interest is new, the node adds it to its interest cache. If the interest already exists in the cache but there is no gradient for the sender of the interest, a gradient is added with the requested data rate to the interest entry. Finally, if both the interest and the gradient already exist, only the lifetime of the gradient is updated. When a lifetime of a gradient field expires, it is removed. Correspondingly, the whole interest entry is removed when there are no gradient fields left. Each node further rebroadcasts all interests that it receives for the first time. In practice, interests are flooded to the whole network and at the same time each pair of neighboring nodes establishes a gradient toward each other.

When a node within the area specified in the interest receives the interest, it starts sensing the area and generating event samples at the rate defined in the interest. A data packet corresponding to an event sample is sent to all neighbors to which there is a gradient entry. A node that receives a data packet compares the attribute-value pairs of the data packet to entries in the interest cache. If no matching entry is found, the packet is dropped. For a matching entry, the lifetime of the gradient is updated. Nodes also keep a data cache of received data packets in order to suppress copies of the same data packet and to potentially aggregate data before resending it.

After the sink receives the first data packet, it starts periodically sending reinforcement interests to a certain neighbor selected by data driven local rules. The local forwarding rule can be simply to send a reinforcement to the neighbor from which the latest event matching the interest was received. When each intermediate node uses the same rule, a path between the sink and the sensed area is formed. Recall that the unused gradients are removed because each gradient field has a certain lifetime. Thus, the sink periodically reinforces the best paths and others eventually time out. In addition, the reinforcements can adaptively modify the data-sending rate from the sensed area by setting new requested data rate to reinforcements.

Directed diffusion scales well since the number of gradients kept by the nodes depends on the number of different interest queries and on the node density, not on the number of nodes in sensed areas (sources) or on the number of sinks [19]. Thus, it is suitable for large sensor networks with a reasonable number of different interests.

SPIN is designed for sensor network applications that involve gathering data about multiple phenomena or about multiple aspects of a single phenomenon. SPIN disseminates data about a certain observation to all nodes that are interested in the data, treating all interested nodes as potential sink nodes. The idea of SPIN is to describe each data packet with a short meta-data descriptor and to use these meta-data descriptors when nodes are negotiating about the transmission of data. Note that using the meta-data descriptions is basically the same idea as using attribute-value pairs to describe data in directed diffusion.

The basic operation of SPIN is rather simple. When a node has data to share, it broadcasts an ADV packet containing meta-data to all its neighbors. All interested neighbors respond with a REQ packet containing the same meta-data. The node can then send a DATA packet that has the meta-data as a header to all interested neighbors. When the same operation is repeated, the data eventually disseminates to all interested nodes. The authors of [79] also proposed modifications to the basic method, which take into account the remaining energy resources at a node and the existence of asymmetric and lossy links.

SPIN is useful in applications that require the flooding of information to large parts of a network. It is more efficient than normal flooding since only interested nodes receive the data and nodes can avoid receiving multiple copies of the same data by not sending REQs for them. However, even if the network is connected, SPIN cannot guarantee the delivery of data to all interested nodes. Delivery failures can happen when nodes between a source and a group of potential sinks are not interested in the data. If the only possible paths from the source to the potential sinks go through those nodes, the data will not be delivered.

4.6.2 Minimum energy topology control for location-based routing

The problem of forming a Small Minimum-Energy Communication Network (SMECN) is studied in [80]. It is assumed that each node can adjust its transmission power unrestricted between zero and a predefined maximum value. If the network in which each node transmits with its maximum power is repre-

sented by a network graph $G_{\max}$, the corresponding SMECN G satisfies:

1. G includes all the nodes of $G_{\max}$ but has fewer edges,
2. all nodes that are connected in $G_{\max}$ are also connected in G and
3. G has the minimum-energy property.

The minimum-energy property guarantees that for every connected pair of nodes (u, v) in $G_{\max}$, G includes a path between u and v that consumes the minimum amount of energy among all possible paths. The amount of consumed power in path $r = (u_0, ..., u_k)$ is

$$C(r) = kD + \sum_{i=0}^{k-1} K \cdot l(u_i, u_{i+1})^{\alpha}, \tag{4.1}$$

where k is the number of hops in the path, D is the power consumed when receiving a packet and $K \cdot l(u_i, u_{i+1})^{\alpha}$ represents the path loss as given by (3.1) in transmission from node u_i to u_{i+1}.

It was proven in [80] that a subgraph G is a SMECN if all 2-redundant edges are removed. Edge (u, v) is k-redundant is there is a path r in $G_{\max}$ such that $|r| = k$ and $C(r) \leq C(u, v)$. The removal of 2-redundant edges can be done if each node first broadcasts its location with the maximum power. Then each node u removes its 2-redundant edges using the simple algorithm:

1. For each neighbor v, calculate $C(u, v)$.
2. Remove edge (u, v) if for any other neighbor w, $C(u, w, v) \leq C(u, v)$.

Figure 4.17 shows network graph $G_{\max}$ and its corresponding SMECN G with 2-redundant edges removed when $D = 0$, $K = 1$ and $\alpha = 4$. Note that for parameters $D = 0$, $K = 1$ and $\alpha = 2$, the SMECN equals the Gabriel graph presented in Section 4.5.2.

The authors in [80] point out that broadcasting using maximum power potentially wastes energy since in dense networks, it may require much less transmit power to prune the 2-redundant edges. For dense networks, an alternative algorithm was proposed that increases the power gradually until it is not possible that 2-redundant edges still exist. The details of that algorithm are not presented here. Once each node has locally computed its SMECN edges, the SMECN can be used, for example, for geographic greedy forwarding. Depending on the interaction between MAC and forwarding, each node can adjust its transmission power for each edge independently or the power that is needed for reaching the farthermost neighbor can be used in all transmissions.

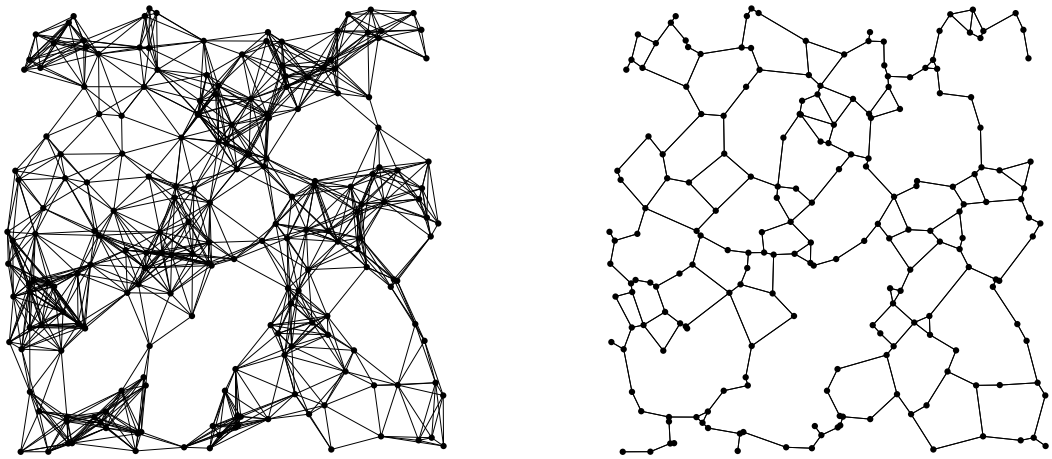


Figure 4.17: A network graph $G_{\max}$ and is corresponding SMECN G with 2-redundant edges removed.

Note that it is also possible to find a SMECN $G_{\min}$ that has the smallest possible number of edges. However in multihop networks, the calculation of $G_{\min}$ cannot be done locally because it requires knowledge of topology beyond the maximum transmission range of a node. An algorithm for finding $G_{\min}$ in a full mesh network is presented in [81].

Chapter 5

Performance study of some geographic forwarding methods

In this chapter, we define a model for analysing and optimizing the performance of geographic forwarding methods. Next, the geographic forwarding methods used in this study are presented. Finally, we discuss previous work that is related to our study.

5.1 Introduction

We consider a multihop ad hoc network such that the overall number of nodes in the network is large. Because of the large number of nodes, a typical distance between a randomly selected source-destination pair is much greater than a typical distance between neighboring nodes and a typical path in the network consists of a large number of hops.

A network defined with the above assumptions can be analyzed in a macroscopic and microscopic level [82]. At the macroscopic level, routes between a source and a destination are smooth curves and the underlying network can be considered as a fabric forming a homogenous, continuous medium. The used routing strategy defines whether the routes are just straight lines corresponding to the shortest-path routing or general smooth curves corresponding to some other routing metrics. This is related to network layer functionality only. Routing along a predefined smooth curve in a dense ad hoc network is also considered in [83]. The microscopic level considers the network from a single node's point of view. At this level, each node makes a forwarding decision locally depending only on the direction in which a packet is traversing. This involves functionality both from network and link layers, implying a need

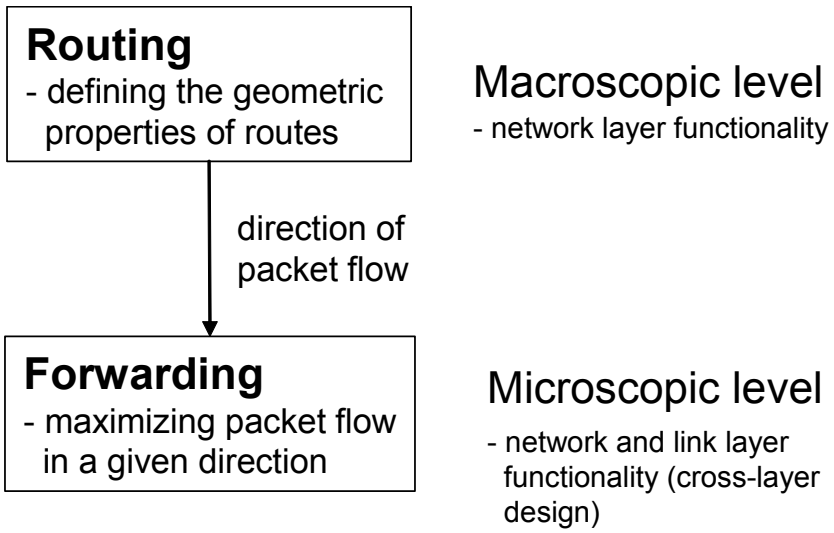


Figure 5.1: The problem decomposition presenting the tasks of the macroscopic and microscopic levels and the connection between them.

for cross-layer solutions. If directions can be treated as being independent, the problem is to maximize the flow of packets in a given direction. Independence of the directions can be achieved, e.g., by using scheduling at the macroscopic level to assign certain time shares for the different directions. This problem decomposition is illustrated in Figure 5.1.

In this study, we focus on the forwarding problem at the microscopic level. The flow of packets at the microscopic level depends on the MAC protocol, the forwarding rules and the node density. To simplify the MAC part, we assume the use of slotted ALOHA, which is characterized by a single parameter defining the probability to transmit in a given time slot. Thus, our aim is to find the maximum throughput of the network by maximizing the mean flow of packets in a given direction with respect to the slotted ALOHA parameter (the transmission probability) and network density for alternative local forwarding rules.

5.2 Network model and assumptions

The locations of nodes in the network are assumed to be distributed according to the two-dimensional Poisson point process with intensity λ [1/m^2]. The network topology is static, and thus node mobility or failures are not considered in this study. Each node transmits with the same power resulting in a transmission range with radius R [m] and the simple, commonly used Boolean

model presented in Section 3.2 is used to model the effect of interference. In addition to the simple interference model, we fix the packet size and use slotted ALOHA as a MAC protocol. The operation of slotted ALOHA was presented in Section 3.3. We assume that ACK packets are much smaller than data packets. Thus, the time slot duration is dominated by the transmission time of data packets.

It is assumed that in addition to its own coordinates, each node also knows the coordinates of its neighbors. A node can receive its coordinates from the GPS system and if it and all its neighbors initially broadcast the coordinates to all neighbors, the neighborhood-wide knowledge of the locations is achieved. Using the local location information and the direction in which a packet is traversing, each node can make the forwarding decision locally.

The mean flow of packets is characterized by the packet flow intensity, which is defined as the number of packets crossing a line of unit length perpendicular to the direction of the packet flow in a time unit. Using this definition the mean packet flow intensity I can be expressed as

$$I = \rho \mathbf{v_x} \qquad \left[\frac{1}{\mathrm{m} \cdot \mathrm{s}}\right], \tag{5.1}$$

where ρ is the packet density $[1/\mathrm{m}^2]$ and $\mathbf{v_x}$ is the average packet velocity projected to the direction of the packet flow [m/s].

An alternative way of presenting the mean packet flow intensity is to use the mean density of progress that was already defined in [55]. The mean density of progress is defined as the average progress of packets per time slot per node. Let $N_R = \lambda \pi R^2$ be the average number of nodes within the transmission range and p be the transmission probability of slotted ALOHA. Later for brevity, we will refer to N_R as the network density. Note that if the network is considered as a graph, N_R is the average degree of a node. The mean density of progress is given by

$$I = \frac{\sqrt{\lambda}}{t} \cdot u(N_R, p) \qquad \left[\frac{1}{\mathrm{m} \cdot \mathrm{s}}\right], \tag{5.2}$$

where t is the time slot length [s] and $u(N_R, p)$ is the average dimensionless progress of packets per time slot per node. In order to obtain a dimensionless quantity, progress has to be measured using a unit length related to the network model. In our network model, there are two possible length quantities: the transmission radius R and $1/\sqrt{\lambda}$. We chose to use the latter, $1/\sqrt{\lambda}$, as a unit length of distance for progress. It was shown in [55] that $1/(2\sqrt{\lambda})$ is the average distance between two nearest nodes. Thus, $u(N_R, p)$ measures the progress in terms of the mean number of nearest neighbor distances multiplied by a factor 1/2. Finally, in order to obtain the mean density of progress I with dimensions, $u(N_R, p)$ is multiplied with $\sqrt{\lambda}/t$.

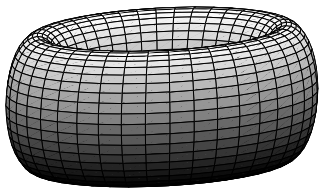


Figure 5.2: An example of a torus

5.3 Simulation model

We maximize $u(N_R, p)$ via simulations. To this end, we simulate a finite area representing a snap shot of our assumed large (infinite) network. To keep simulation work manageable, we try to keep the dimensions of the area as small as possible relative to the node density. However, this introduces the problem of border effects: nodes near the border of the network see different traffic and interference patterns than nodes in the middle of the network. To avoid harmful border effects, we seam the opposite sides of the plane together. The result is a torus that has a surface area equal to the area of the original plane. An example of a torus is shown in Figure 5.2. Because of the toroidal shape of our network, a packet sent over the border of the network is transmitted along a route at the opposite side of the network.

Let us consider a square plane (or a surface of a torus) with a unit area. The locations of nodes are drawn according to the two-dimensional Poisson point process as follows:

1. Draw the number of nodes N from the Poisson distribution, $N \sim \text{Poisson}(\lambda)$.

2. For each node i, $i = 1, 2, \ldots, N$, draw the x-coordinate from the uniform distribution $X_i \sim \text{U}(0, 1)$ and the y-coordinate from the uniform distribution $Y_i \sim \text{U}(0, 1)$.

Traffic is generated to the network by initially placing the same number of packets M in each node. The initial number of packets is chosen large enough that a further increase in packets would have no substantial effect on the density of progress. On the other hand, the initial number of packets is set such that the heavy traffic assumption would be valid if traffic was evenly spread over the network. The determination of M is discussed in more detail later in Section 6.1.2. The heavy traffic assumption is common in analytic ad

hoc network performance studies and it says that every node has a packet to send all the time. The packets have a lifetime equal to a simulation length and no new packets are generated during the simulation. The direction of the packet flow is fixed to be from west to east.

In our simulation setting λ and t are fixed, so the task of maximizing I becomes a task of maximizing $u(N_R, p)$ instead. During the simulations, $u(N_R, p)$ is calculated using

$$u(N_R, p) = \frac{\sum S_i}{TN} \cdot \sqrt{\lambda}, \tag{5.3}$$

where S_i is the total progress of packet i and T is the number of time slots. Recall that, (5.3) measures the progress in terms of the mean number of nearest neighbor distances (multiplied by a factor $1/2$). Thus, the task is to find such N_R and p that maximize $u(N_R, p)$ for a given local forwarding method. Different forwarding methods can then be compared using the maximum $u(N_R, p)$ as a performance measure.

5.4 Forwarding methods

Four different forwarding methods are compared in our simulations. The first is a deterministic greedy forwarding algorithm, the second and the third algorithms attempt to spread traffic by randomizing the choice of a next hop and the fourth algorithm opportunistically chooses the best available next hop. Routing around concave nodes is handled similarly in all algorithms using the greedy/flooding method described in 4.5.2. However, for our static network, it is not very sensible to declare concavity for each packet separately. Instead, each node that notices its concavity with respect of the direction of the packet flow declares its concavity to all neighbors and refuses to receive any further packets. In effect, greedy/flooding recursively removes all concave nodes from the network.

5.4.1 Most forward within radius

The operation of MFR was already described in Section 4.5.1. We have implemented MFR without the option to choose a node with the least negative progress as a next hop, i.e., negative progress is not allowed due to formation of routing loops. The choice of a next hop is simple in our network model. Because each packet is traversing from west to east, the next hop is simply the neighbor with the greatest x-coordinate. However, the nodes located near the eastern border of the area have to be considered differently. Because of

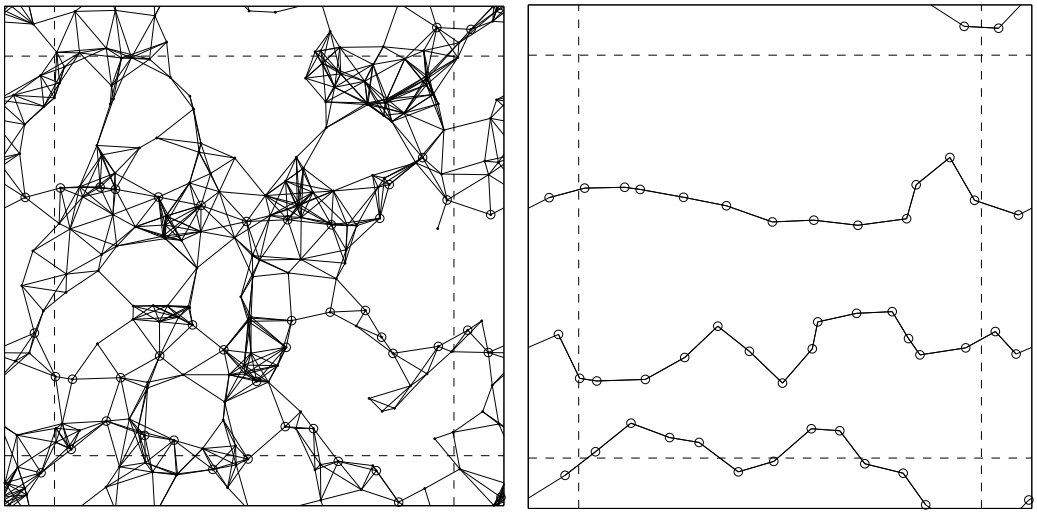


Figure 5.3: A network and its 3 MFR paths. The border areas from which it is possible to have a link to the opposite side of the network are marked with dashed lines.

the toroidal shape of the area, their most forward neighbors can also be on the opposite side of the network.

In a static multihop network, where each packet is traversing in the same direction, packets tend to flow along certain paths. This is easy to see in our torus model in which no new packets are generated. After the initial transition, all packets traverse along a few paths called MFR paths. We define an MFR path as a cycle around the toroidal network, where each next hop is chosen using the MFR rule. See Figure 5.3 for an example of a toroidal network and its MFR paths. Note that it is also possible that the same MFR path goes around the torus for multiple rounds before returning to a starting node. Figure 5.4 shows an example of an MFR path going around the torus for three rounds.

If Figure 5.3 is examined carefully, it can be noticed that there exists a certain analogy between the routes of packets and an aquatic system consisting of brooks and rivers. All packets initially located in a node that is not along an MFR path follow the same brook until it joins a river (an MFR path). Packets once forwarded into a river stay there until the river reaches a sea (a destination) infinitely far away. As can be seen from Figure 5.3, the network utilization is low. It will be seen in Chapter 6 that this results in poor mean density of progress.

If there exists at least one path around the network and concave nodes have been recursively removed, MFR paths can be found using the following algo-

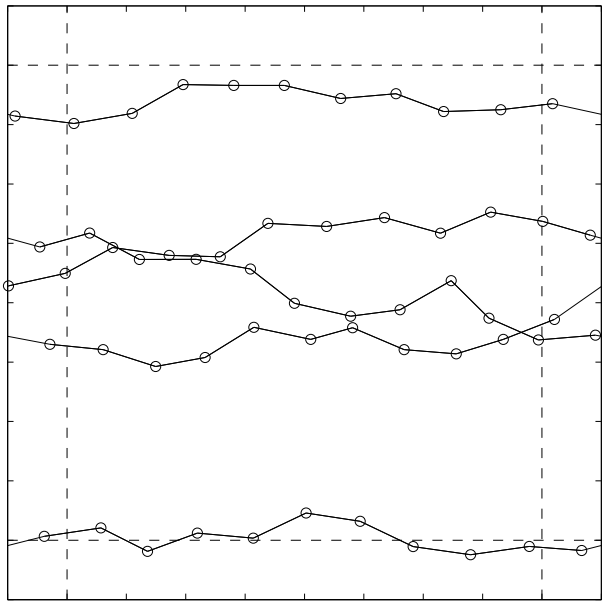


Figure 5.4: A network with only MFR paths shown. The border areas from which it is possible to have a link to the opposite side of the network are marked with dashed lines.

rithm:

1. Set $i = 1$.
2. Set $j = i$.
3. If node j is visited, go to step 8.
4. Mark node j visited.
5. If the most forward neighbor k of the node j is not visited, set $j = k$. Go to step 4.
6. If the most forward neighbor k of the node j is not active or passive, mark node k active and set $j = k$. Go to step 4.
7. Set each node passive, which was visited during iteration i and is not active.
8. Set $i = i + 1$.
9. If $i < N$, go to step 2. Otherwise, stop.

The nodes marked active belong to an MFR path and the nodes marked passive act as access nodes forwarding packets towards MFR paths. The algorithm selects a node and starts to traverse the network always choosing an MFR neighbor. As the algorithm approaches to a visited node, it checks whether it is already marked as active or passive. If the node is not active or passive, it must belong to a new MFR path. The new MFR path is completely traversed marking all nodes along it active. When the algorithm reaches the first node that is marked active or passive, it stops traversing and marks all non-active nodes that were visited during the traversal passive.

5.4.2 Random forwarding

The easiest way to arrange traffic load spreading at the microscopic network level is to randomize the local forwarding decision. In random forwarding, we simply set an equal probability for each forward neighbor to become chosen as a next hop. Random forwarding spreads the packet flow effectively over the whole network. The cost of better utilization of the network is that some of the hops are very short.

5.4.3 Weighted random forwarding

The idea of weighted random forwarding (WRF) is to increase the average hop length of random forwarding by weighting the next hop probabilities depending on the locations of the neighbors. In weighted random forwarding, the probability q_{ij} that sending node i chooses a forward neighbor j as a next hop is

$$q_{ij} = \frac{l(i,j)}{\sum_{k=1}^{N_F} l(i,k)}, \tag{5.4}$$

where N_F is the number of all forward neighbors.

5.4.4 Opportunistic forwarding

If the MAC protocol assumption is loosened, it is possible to implement a forwarding algorithm that opportunistically chooses the best available next hop. Opportunistic forwarding is a slightly modified version of the ExOR protocol presented in Section 4.5.3. The operation of opportunistic forwarding during a time slot goes as follows:

1. At the beginning of a time slot, a sender broadcasts a packet to all its

forward neighbors with probability p. The sender includes a forward neighbor list that is prioritized by progress into the packet.

2. All forward neighbors not hearing another transmission receive the packet. Each forward neighbor prepares to send an acknowledgement after a delay proportional to its position in the forward neighbor list.

3. Acknowledgements are sent. A forward neighbor hearing an acknowledgment with a higher priority than its own cancels its own scheduled acknowledgement. The sender hears at least one acknowledgment.

4. The sender sends a permission to send (PTS) packet to the neighbor from which it received the highest priority acknowledgement.

5. The neighbor that received the PTS stores the packet in its queue. Other forward neighbors drop the packet.

In contrast to ExOR in opportunistic forwarding, a sender makes the decision of a next hop based on received acknowledgements. The acknowledgement scheme of ExOR is not used because it cannot guarantee the absence of duplicate packets. The acknowledgement scheme of opportunistic forwarding increases the amount of control traffic compared to ExOR and the basic slotted ALOHA. However, the increased control traffic has no significant effect on the time slot duration since the network model includes the assumption that the time slot duration is dominated by the transmission time of data packets.

Because the medium access in opportunistic forwarding does not strictly follow the slotted ALOHA protocol, the mean density of progress of opportunistic forwarding is not directly comparable to three previous algorithms. Nevertheless, opportunistic forwarding gives a good approximation of the upper limit performance in our network model for a contention-based MAC scheme.

5.5 Related work

One of the earliest multihop performance analysis of slotted ALOHA was presented by Takagi and Kleinrock [55]. The goal of the study was to find the optimal number of nodes within a transmission range N_R^* and the corresponding optimal transmission probability p^*. Both the optimization of the average number of successful transmissions per time slot per node $s(N_R, p)$ and the dimensionless (scaled with $1/\sqrt{\lambda}$) mean density of progress $u(N_R, p)$ were considered. The network model was very similar to our model: all nodes transmit with the same power, the Boolean interference model is used, nodes are distributed according to the two-dimensional Poisson process and MFR is used

for forwarding. The difference is that they assumed that all nodes have always packets to send (heavy traffic assumption), which clearly is not true in our network because of the formation of MFR paths.

According to the analysis, both $s(N_R, p)$ and $u(N_R, p)$ are maximized by

$$p^* = \frac{2}{N_R + 2 + \sqrt{N_R^2 + 4}} \tag{5.5}$$

for a given value of N_R. Using p^* given in Equation 5.5, it was found that $u(N_R, p)$ is maximized when $N_R \approx 7.72$. So, the mean density of progress is maximized when there are on the average $N_R^* \approx 7.72$ nodes within a transmission range R and when each node transmits with probability $p^* \approx 0.113$.

The problem of finding the optimal slotted ALOHA transmission probability that maximizes the mean density of progress is also considered in [84]. Interference is modelled with the more realistic Physical Model [20] presented in Section 3.2 and the transmission powers are exponentially distributed independently between nodes and time slots. It was shown that if the Physical Model with background noise $N = 0$ is assumed, the optimal transmission probability p^* does not depend on the node density λ. The authors compared CSMA to slotted ALOHA and pointed out that for optimal density of progress, the carrier sense range of CSMA have to be adapted to the node density.

In [85], the same network model is used as in [84], only the assumption of exponentially distributed transmission powers is modified. Instead, when all nodes are assumed to transmit with the same constant power, the distribution of the probability of successful reception in a random time slot is evaluated. The first result is an approximation for the probability of successful reception averaged over all surrounding node configurations. The latter part presents a recursive method for evaluating the distribution of the temporal probability of successful reception over different surrounding node configurations. If computational effort is increased, both approximations can be made arbitrary accurate.

Our idea of considering a large ad hoc network at the macroscopic and microscopic level is taken from [82]. The authors study a dense wireless multihop network at the macroscopic level and present a general framework for analyzing load balancing. The load balancing problem is to determine routes such that the maximum scalar flux is minimized. The scalar flux of packets is dependent on the MAC protocol and it is defined to be the rate at which packets enter a disk with diameter d at point $\mathbf{r}$ divided by d in the limit when $d \to 0$. The main results of the study are lower bounds for the maximum scalar flux and a general expression for determining the packet flux at a given point for a given set of curvilinear paths.

Chapter 6

Results

This chapter summarizes the results obtained by simulating a network described in the previous chapter. First, we present some of the issues we encountered during simulations. Second, the results from maximizing the mean density of progress as well as maps of the traffic distribution are presented.

6.1 Practical simulation issues

There were some practical simulation issues that had to considered before the actual simulations were started. This section summarizes the most important of them and how they were taken into account in our study.

6.1.1 Removal of concave nodes

As it was already stated in Section 5.4, all four forwarding algorithms disconnect concave nodes from the rest of the network. This is implemented by recursively removing concave nodes until there are no concave nodes left. Figure 6.1 depicts the percentage of recursively removed concave nodes as a function of N_R when $\lambda = 1000$ [1/unit area] (recall that the surface of the torus corresponds to a square plane with a unit area). For each value of N_R, 1000 different node location realizations were created. When there exists no path around the torus, all nodes are recursively removed. Note that for dense networks ($N_R > 10$), which is the case we are studying, it is common that there are only a few or no concave nodes.

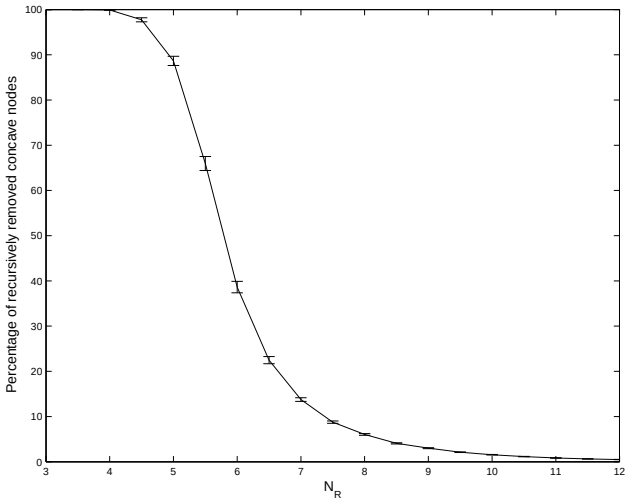
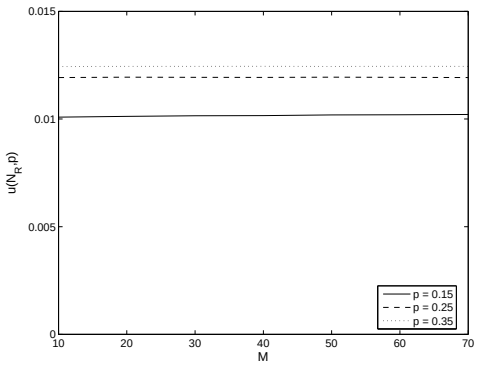


Figure 6.1: The percentage of recursively removed concave nodes as a function of N_R with $\lambda = 1000$ [1/unit area]. The 95% confidence intervals are shown as error bars.

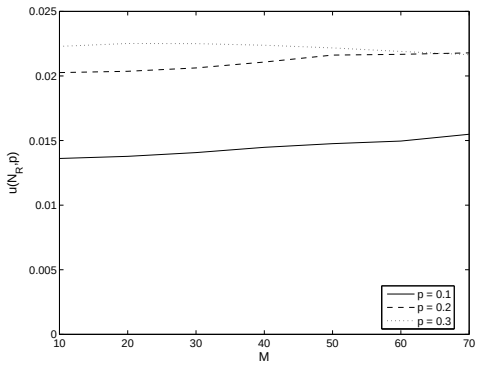
6.1.2 Number of packets

Initially, the same number of packets, M, were placed in each active node. Recall that an active node is a node that participates in the relaying of traffic in a given direction. In random forwarding, WRF and opportunistic forwarding, the set of active nodes consists of those nodes that are left after the removal of concave nodes. In MFR, the set of active nodes is the set of nodes belonging to MFR paths, which can be determined by using the algorithm given in Section 5.4.1. By placing packets only in active nodes, the initial transient duration of a simulation can be reduced.

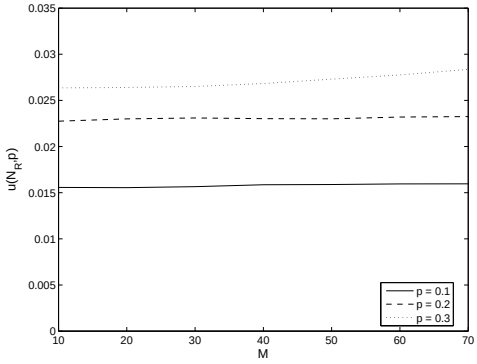
According to our idea, the network is simulated under heavy traffic. Thus, M has to be chosen large enough such that a further increase would have no significant effect on $u(N_R, p)$. In Figure 6.2, the dimensionless mean density of progress $u(N_R, p)$ is drawn as a function of M for a few values of p, for typical N_R and for a given forwarding algorithm. In each figure, the same random node location realization is used for each value of p. Since all the curves in Figure 6.2 are rather flat at $M = 50$, we chose the value of $M = 50$ to be used with all forwarding algorithms.



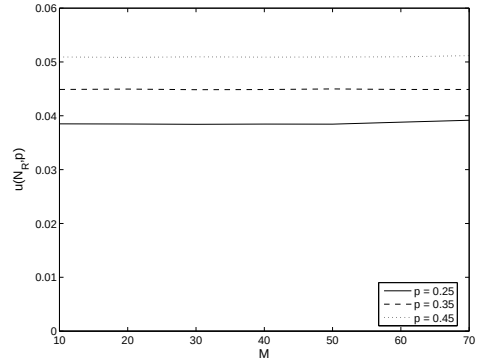
(a) MFR, $N_R = 50$



(b) Random, $N_R = 14$

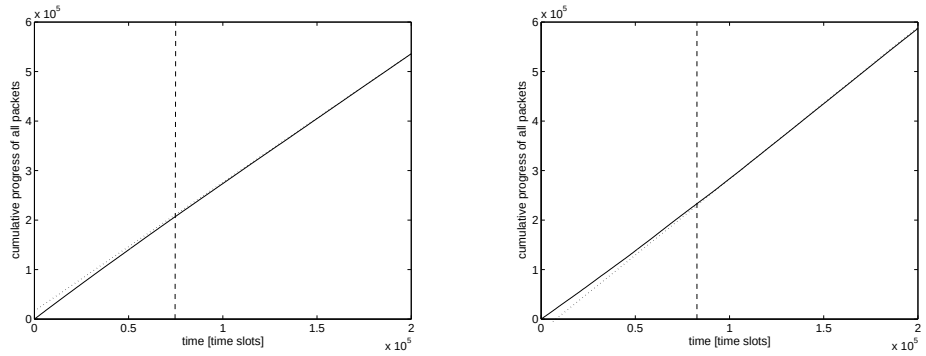


(c) WRF, $N_R = 14$



(d) Opportunistic, $N_R = 18$

Figure 6.2: The dimensionless mean density of progress $u(N_R, p)$ as a function of M.



(a) $N_R = 20, p = 0.15$ (b) $N_R = 70, p = 0.45$

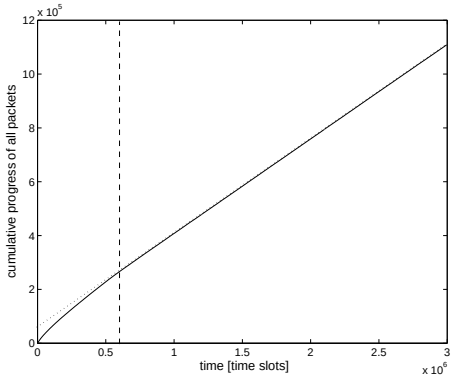
Figure 6.3: The cumulative progress of all packets as a function of time for MFR with $\lambda = 9000$ [1/unit area]. The point at which the slope of the curve is stabilized is graphically estimated and it is marked with a vertical dashed line.

6.1.3 Initial transient duration

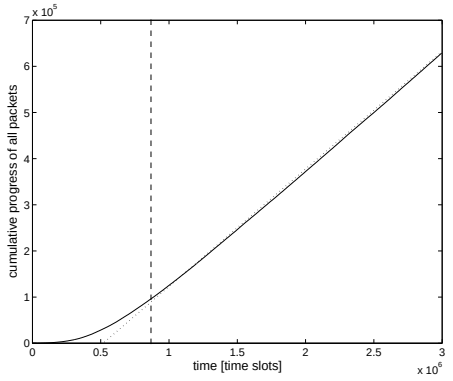
In this study, we are interested in the steady state of the network. Generally, the steady state is reached when the impact of the initial state of a system becomes negligible. In our network model, the steady state is reached when the average queue size in each node has stabilized. Because it would have been awkward to monitor queue sizes of every node individually, we decided to monitor the cumulative progress of all packets instead. If the cumulative progress of all packets is plotted as a function of time, it can be seen that the slope of the curve slowly stabilizes. Because the cumulative progress of all packets grows at a constant pace in the steady state, the stabilization of the curve corresponds to the beginning of the steady state.

We plotted several curves for each forwarding algorithm and for different N_R and p values. Figures 6.3 – 6.6 show the cumulative progress of all packets of a random node location realization for certain parameter values for each forwarding algorithm. The parameter values were chosen near the limits of the parameter range such that the expected initial transient duration is long. Thus, Figures 6.3 – 6.6 represent approximations for the worst-case initial transient duration.

In order to keep computation times feasible, initial transient durations that are used in simulations should be as short as possible. Combining this principle with the results from the transient curves, we chose a suitable initial transient duration to be used in simulations for each forwarding algorithm. The initial transient durations are shown in Table 6.1.

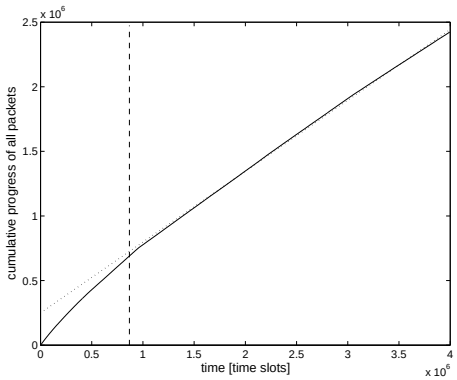


(a) $N_R = 10, p = 0.1$

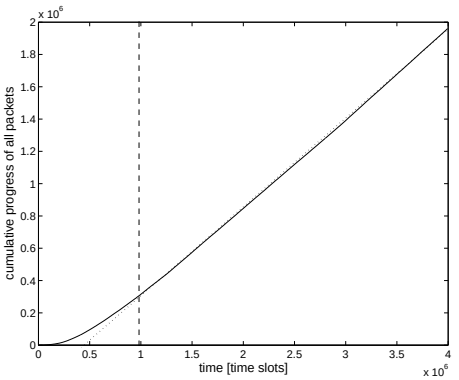


(b) $N_R = 18, p = 0.45$

Figure 6.4: The cumulative progress of all packets as a function of time for random forwarding with $\lambda = 1000$ [1/unit area]. The point at which the slope of the curve is stabilized is graphically estimated and it is marked with a vertical dashed line.

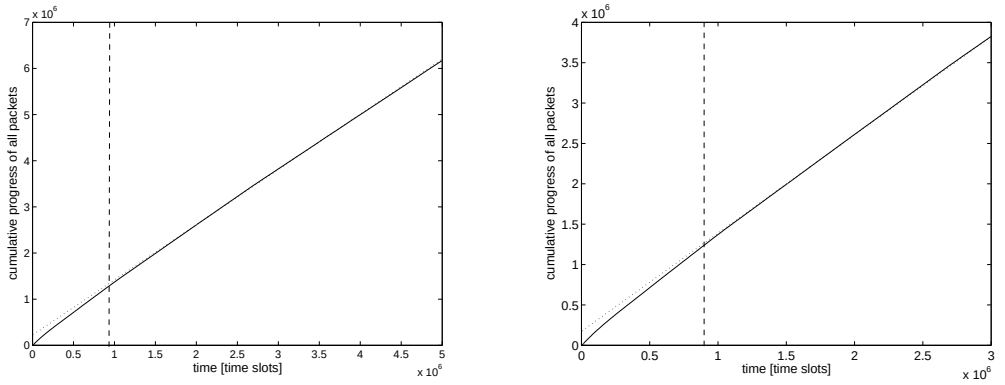


(a) $N_R = 10, p = 0.15$



(b) $N_R = 18, p = 0.4$

Figure 6.5: The cumulative progress of all packets as a function of time for WRF with $\lambda = 1000$ [1/unit area]. The point at which the slope of the curve is stabilized is graphically estimated and it is marked with a vertical dashed line.



(a) $N_R = 12, p = 0.3$ (b) $N_R = 20, p = 0.5$

Figure 6.6: The cumulative progress of all packets as a function of time for opportunistic forwarding with $\lambda = 1000$ [1/unit area]. The point at which the slope of the curve is stabilized is graphically estimated and it is marked with a vertical dashed line.

Table 6.1: The used initial transient durations for each forwarding algorithm

Forwarding algorithm	Initial transient duration [time slots]
MFR	100000
Random forwarding	600000
WRF	800000
Opportunistic forwarding	1000000

6.1.4 Sufficient number of nodes

The number of nodes needs to be carefully selected to fulfill the assumption of a large network and to keep computation times feasible. The first setting was to position nodes with density $\lambda = 1000$ nodes/(unit area) to the plane. This value fit well in random forwarding, WRF and opportunistic forwarding and they were used in simulations.

In MFR simulations, the average number of nodes $N = 1000$ was too small and caused unexpected variations in the MFR path formation. This is illustrated in Figure 6.7 in which the mean percentage of active nodes averaged over 200 different node location realizations is plotted as a function of N_R for different mean number of nodes. The overall declining trend of the curves flattens and the unexpected variations are decreased when the mean number of nodes is increased.

In practice, we noticed that computation time becomes rapidly infeasible as the

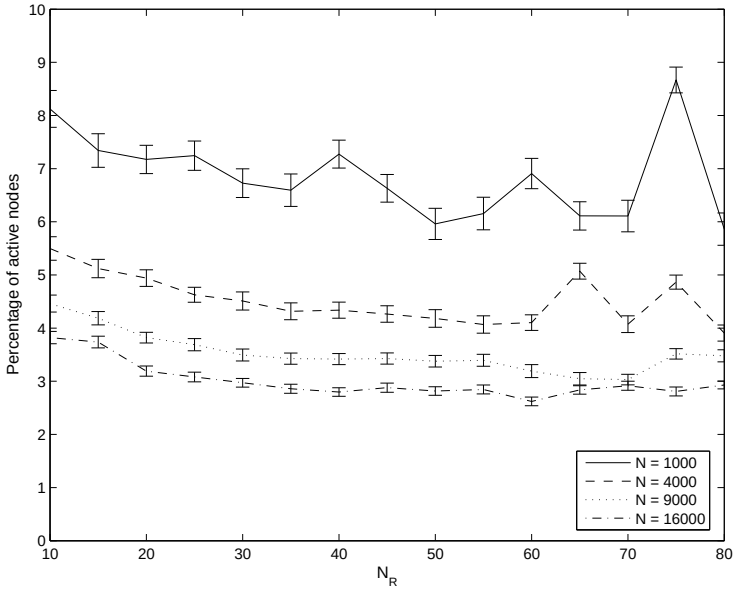


Figure 6.7: The mean percentage of active nodes averaged over 200 different node location realizations as a function of N_R. The 95% confidence intervals are shown as error bars.

average number of nodes is increased. We made a tradeoff between simulation accuracy and computation time by choosing the average number of nodes $N = 9000$ to be used in MFR simulations.

6.2 Optimization of mean density of progress

Recall that the purpose of the simulations is to maximize the dimensionless mean density of progress $u(N_R, p)$ with respect to p and N_R for different local forwarding rules. N_R is varied by keeping λ constant and changing the value of R. The used simulation parameters for each scenario are summarized in Table 6.2.

The dimensionless mean density of progress $u(N_R, p)$ is depicted as a function of p for each forwarding algorithm in Figures 6.8 – 6.11. The results from figures are summarized in Table 6.3 in which the maximum $u(N_R, p)$ and corresponding N_R and p values are collected for each forwarding algorithm.

As can be seen from Table 6.3, MFR achieves the worst $u(N_R, p)$ and the optimal N_R is remarkably greater than with other methods. This is due to the formation of MFR paths that results in low utilization of network resources. Recall from Figure 6.7 that the percentage of active nodes belonging to MFR

Table 6.2: The used parameters for each simulation scenario

	MFR	Random forwarding	WRF	Opportunistic forwarding
λ [nodes/unit area]	9000	1000	1000	1000
M [packets/node]	50	50	50	50
p	[0.15,0.45]	[0.05,0.50]	[0.05,0.50]	[0.25,0.60]
N_R	[20,60]	[10,18]	[10,18]	[12,20]
Transient duration [time slots]	100000	600000	800000	1000000
Total simulation time [time slots]	400000	1200000	1600000	2000000
Number of random node location realizations	200	50	50	50

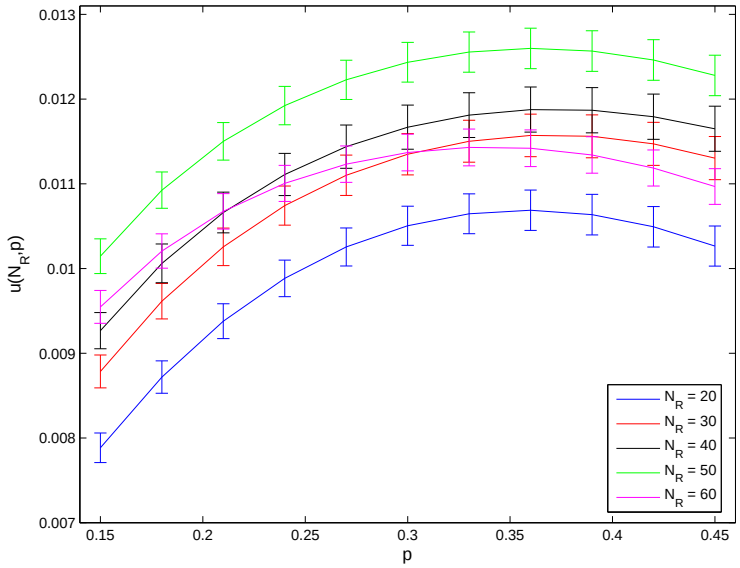


Figure 6.8: The dimensionless mean density of progress $u(N_R, p)$ as a function of p for MFR. The 90% confidence intervals are shown as error bars.

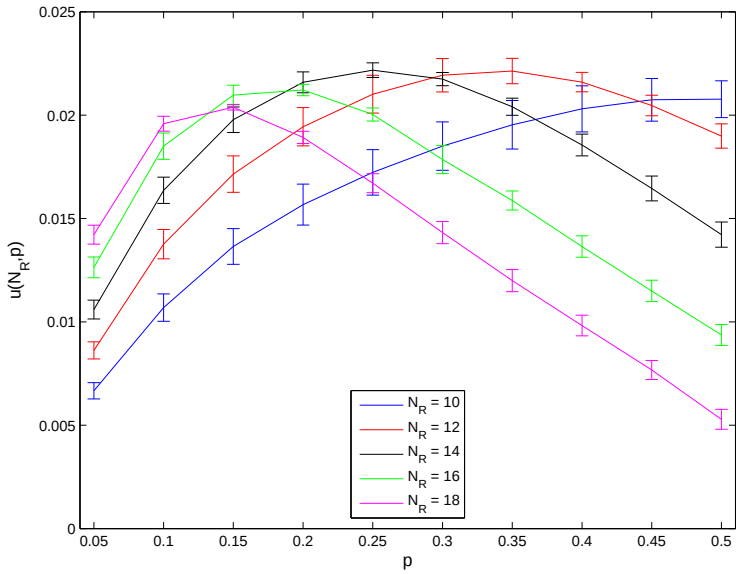


Figure 6.9: The dimensionless mean density of progress $u(N_R, p)$ as a function of p for random forwarding. The 95% confidence intervals are shown as error bars.

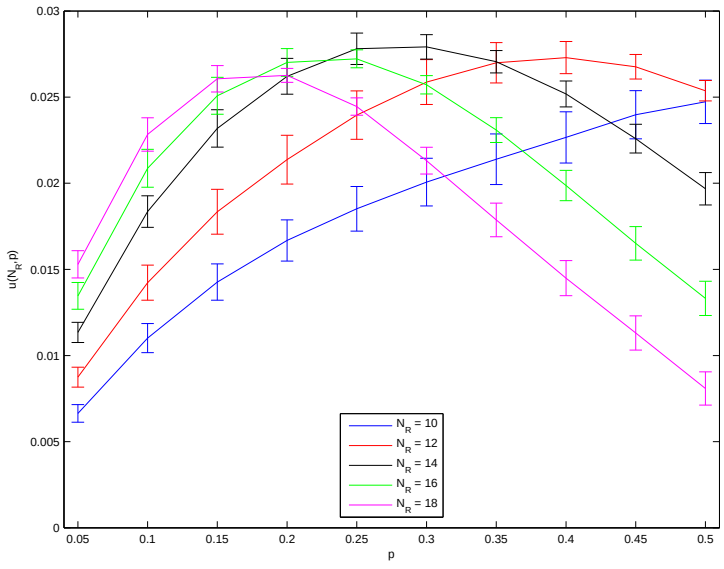


Figure 6.10: The dimensionless mean density of progress $u(N_R, p)$ as a function of p for WRF. The 95% confidence intervals are shown as error bars.

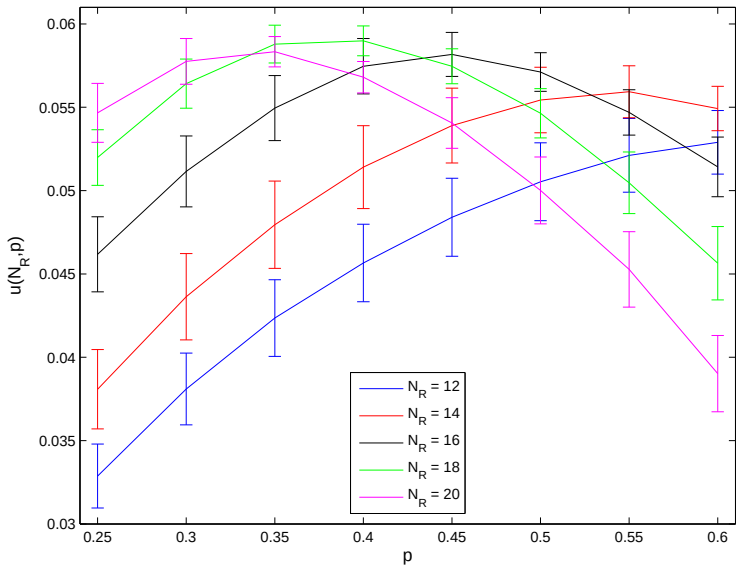


Figure 6.11: The dimensionless mean density of progress $u(N_R, p)$ as a function of p for opportunistic forwarding. The 95% confidence intervals are shown as error bars.

paths is only approximately 3.5 % when $N_R = 50$ and $w = h = 3$. So, there are on the average only $0.035 \cdot 50 = 1.75$ active nodes within a transmission range of R when N_R is optimal.

Random forwarding spreads traffic effectively over the whole network. Better utilization of network resources results in an almost doubled value for $u(N_R, p)$ compared to that of MFR. WRF is able to improve the performance of random forwarding by weighting the next hop probabilities with the relative progress. This results in longer mean hop lengths and better average progress. The curves in Figures 6.9 and 6.10 have very similar shapes, only the optimal p is somewhat higher in WRF. This is reasonable because when WRF is used, there exist nodes that receive only rarely packets to forward. Thus, there is less contention within a transmission range and correspondingly it makes sense to transmit with a higher probability.

As expected, opportunistic forwarding achieves clearly the best $u(N_R, p)$. It should be noted that the performance of opportunistic forwarding is not directly comparable to the other simulated forwarding methods because unlike others, it combines the operation of a slotted MAC protocol with the forwarding method. Opportunistic forwarding always sends a packet to a most forward neighbor that is able to receive the packet; other three forwarding methods

Table 6.3: The maximum $u(N_R, p)$ for each forwarding method.

	$u(N_R, p)$	N_R	p
MFR	0.0126	50	0.35
Random forwarding	0.0222	14	0.25
WRF	0.0279	14	0.3
Opportunistic forwarding	0.0590	18	0.4

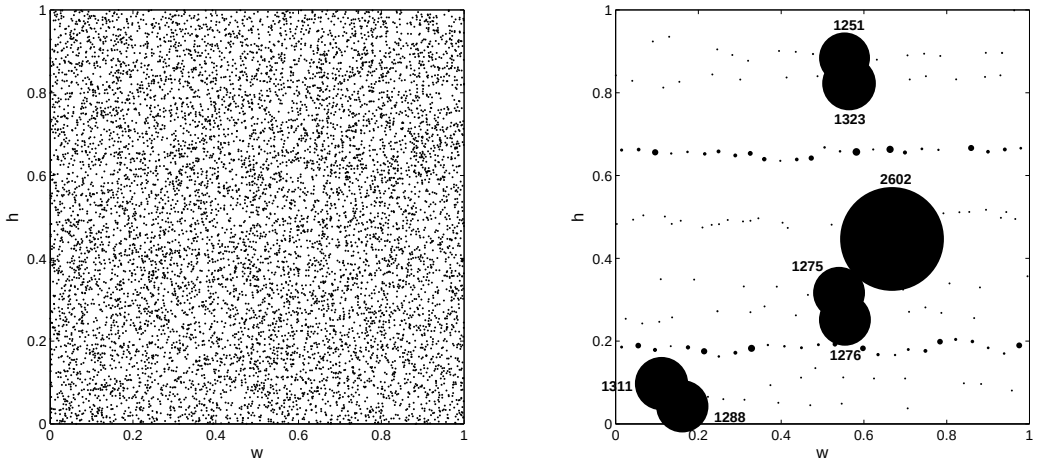
select the next hop node without any information about whether the node is able to receive the packet or not. Because all forward neighbors can potentially receive the packet instead of only one predefined neighbor, collisions are rarer. The collision in opportunistic forwarding corresponds to the situation when none of the forward neighbors is able to receive the packet. That is why the optimal levels of N_R and p are higher than in random forwarding and WRF.

The results in Table 6.3 can be compared to the analytic study of Takagi and Kleinrock [55] presented in Section 5.5. According to their calculations, the maximum dimensionless mean density of progress $u^*(N_R, p)$, which equals 0.0431, is achieved with $N_R^* = 7.72$ and $p^* = 0.113$. Our corresponding simulated values for MFR, random forwarding and WRF are significantly smaller than $u^*(N_R, p)$ because the heavy traffic assumption used in [55] is overly optimistic for our network model. Only opportunistic forwarding reaches greater maximum $u(N_R, p)$ than $u^*(N_R, p)$ due to the different and more efficient MAC scheme. Because there are usually always idle nodes within a neighborhood in our network, it is beneficial to increase the network density (higher N_R) and transmit more aggressively (higher p). Thus, our optimal N_R and p for all forwarding methods are greater than N_R^* and p^*.

6.3 Distribution of packets in the network

To further analyze the ability of the studied forwarding methods to distribute traffic in the network, we have also examined the distribution of packets in the network. These results are illustrated in Figures 6.12 and 6.13, which depict the used node location realization and a snapshot of the instantaneous packet distribution at the end of a simulation. The optimal values of N_R and p are used in each packet distribution figure.

In Figure 6.12b, the radius of a marker at a node is related to the queue size at that node. If a node is not shown, it means that it has no packets at the moment of the snapshot. It should be noted that the same node location realization is not used with MFR as with other forwarding methods because



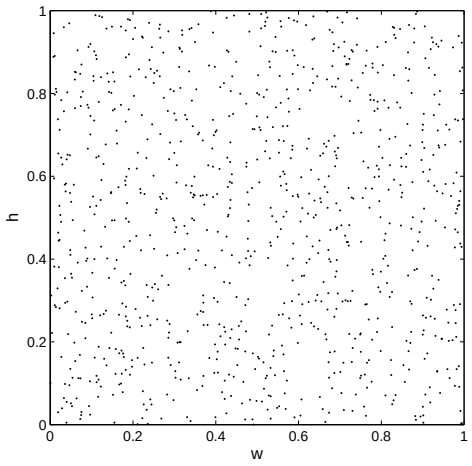
(a) (b)

Figure 6.12: A random node location realization and a snapshot of the packet distribution at the end of a simulation. The marker size in the second figure is related to the queue size at a given node. The number of packets in queues at bottleneck nodes is marked beside the node. $N_R = 50$, $p = 0.35$

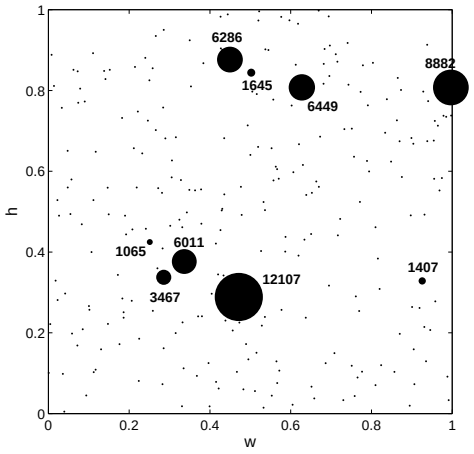
the average number of nodes in MFR simulations is greater ($N = 9000$). As can be seen from Figure 6.12b, MFR utilizes only a small part of the network as packets flow along a few MFR paths. There exist bottleneck nodes whose queue sizes are significantly longer than in other nodes. Bottleneck nodes occur when an MFR path is so close to another MFR path that nodes in different MFR paths cause interference to each other. In Figure 6.12b, there are two MFR paths that are not interfered by other MFR paths. Packets in those two paths are rather evenly distributed among the nodes.

Similar to Figure 6.12, the size of a marker at a node is related to the queue size at that node in Figures 6.13b – 6.13d. Respectively, if a node is not visible, it means that it has no packets at the moment of the snapshot. As illustrated in Figures 6.13b – 6.13d, different nodes become bottlenecks when different forwarding methods are used. Only two nodes are common bottlenecks in all three cases. By examining Figure 6.13, it is hard to conclude why certain nodes become bottlenecks. However, it is clear that bottlenecks exist near the regions in which the network topology is either unusually sparse or dense.

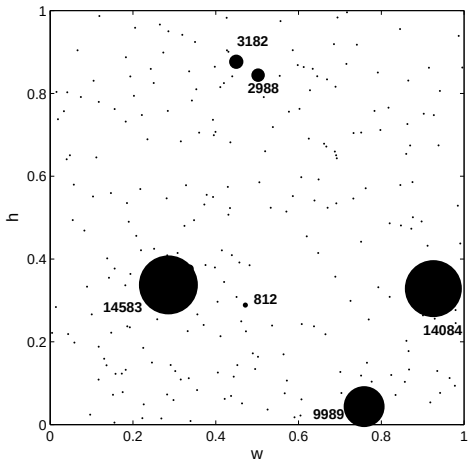
In order to further illustrate the packet distribution in the network, we divided the network with a 10×10 grid and averaged the number of packets in each square over 10 snapshots taken during a simulation. The results are shown in Figure 6.14 with the same node location realization as in Figures 6.12 and 6.13. The shading of a square corresponds to the number of packets in that square



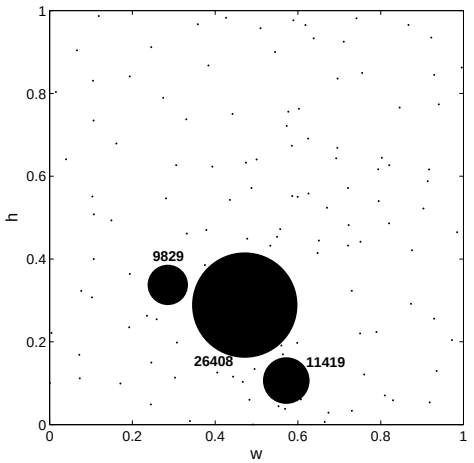
(a) A random node location realization



(b) Random forwarding, $N_R = 14$, $p = 0.25$



(c) WRF, $N_R = 14$, $p = 0.3$

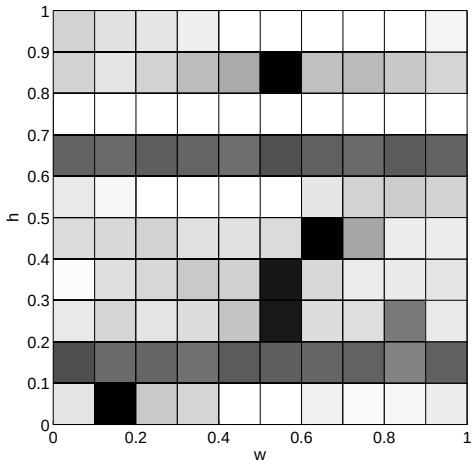


(d) Opportunistic forwarding, $N_R = 18$, $p = 0.4$

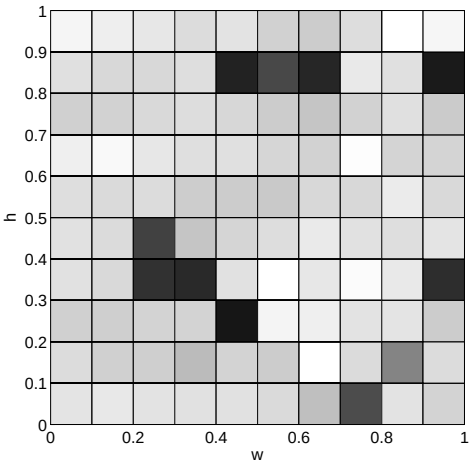
Figure 6.13: A random node location realization and a snapshot of the packet distribution at the end of a simulation for each forwarding method. The marker size in the second figure is related to the queue size at a given node. The number of packets in queues at bottleneck nodes is marked beside the node.

in the logarithmic scale. As can be seen from Figure 6.14, random forwarding and WRF succeed better in traffic spreading than opportunistic forwarding. As traffic distribution figures were generated for other node location realizations, it was found out that in some cases opportunistic forwarding resulted in nearly as good traffic spreading as random forwarding.

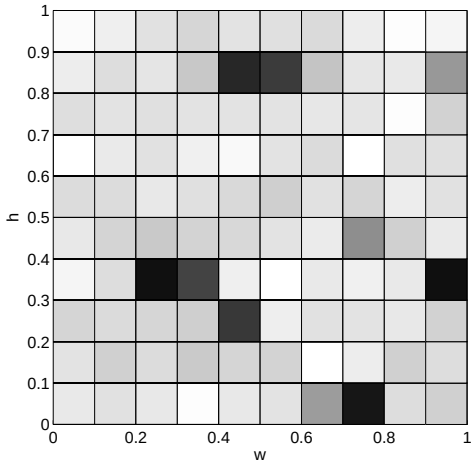
Figures 6.13 and 6.14 give a valuable insight about how serious the bottleneck problem is. If local forwarding rules could avoid the worst bottlenecks, the traffic distribution would be closer to the heavy traffic assumption and the maximum $u(N_R, p)$ would be potentially improved.



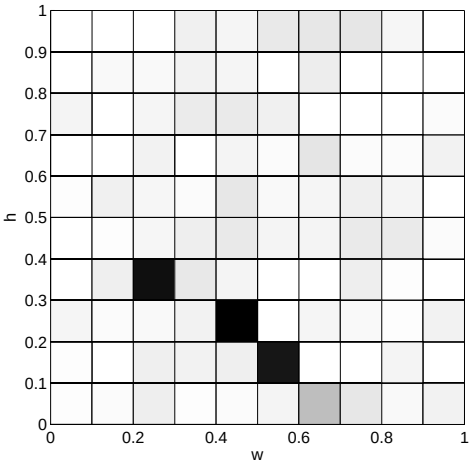
(a) MFR, $N_R = 50$, $p = 0.35$



(b) Random forwarding, $N_R = 14$, $p = 0.25$



(c) WRF, $N_R = 14$, $p = 0.3$



(d) Opportunistic forwarding, $N_R = 18$, $p = 0.4$

Figure 6.14: The number of packets within a square area in the network. The shading of a square corresponds to the number of packets in that square in the logarithmic scale.

Chapter 7

Conclusions

7.1 Summary

An ad hoc network is a wireless network that needs no fixed infrastructure or centralized control in order to operate. Thus, ad hoc networks are especially suitable for applications requiring flexibility and rapid deployment. It is distinctive for ad hoc networks that two nodes can communicate in a multihop fashion. This forces intermediate nodes to forward also relay traffic in addition to their own packets. An ad hoc node has a limited battery capacity and usually batteries cannot be recharged during the operation of the network. Both the multihop nature of communications and limited energy resources have to be considered when designing protocols for ad hoc networks.

In the first part of the study, we made a literature survey of MAC and routing protocols designed for ad hoc networks. Due to the lack of centralized control, most MAC protocols are based on random access in which multiple nodes compete for the access to a channel. A random access MAC protocol should guarantee that the competition for the access is fair and conflicts among the nodes are resolved. In order to decrease destructive interference between nodes, most MAC protocols reserve the channel during a transmission using control traffic. However, reservation-based protocols often fail to completely reserve the channel within a transmission range (the hidden terminal problem) or unnecessarily prevent other transmissions outside the transmission range (the exposed terminal problem). There have been recent proposals to improve MAC performance by using directional antennas and multichannel receivers. In addition to MAC protocols employing that kind of advanced hardware, energy-efficient MAC protocols were also discussed for both ad hoc and wireless sensor networks.

An ad hoc network is a challenging environment for routing. First, node mobility and power saving periods cause rapid topological changes. Second, the bandwidth of the channel is usually very limited and thus the routing overhead should be kept low. Finally, because ad hoc networks in general and especially wireless sensor networks can consist of a large number of nodes, routing protocols have to be scalable. Traditional proactive and reactive routing protocols often fail to scale well in large networks. Significantly better scalability can be achieved in routing if geographic location information is readily available. A complete geographic routing protocol consists of three parts: a local forwarding method, a recovery method to route around concave nodes and a location service protocol to provide the location of a destination to a source node. A promising new approach is to combine geographic forwarding with a locally synchronized MAC scheme. The potential of this approach was also confirmed in the simulation part of the study.

The second part of the thesis is a simulation study to compare the performance of different geographic forwarding methods in a large ad hoc network. A large ad hoc network can be analyzed at a macroscopic and microscopic level. At the macroscopic level, the network can be considered as a homogenous, continuous medium where routes are arbitrary smooth curves. The microscopic level considers the network from a single node's point of view. At this level, a local forwarding decision depends on the direction that is given from the macroscopic level. In our study, we focused on the forwarding problem at the microscopic level assuming that slotted ALOHA is used as a MAC protocol. More precisely, the problem was to maximize the flow of packets measured with the mean density of progress in a given direction with respect to the slotted ALOHA transmission probability and network density for different forwarding rules.

A model of a large network was formed by placing nodes according to the two-dimensional Poisson point process onto a toroidal area. The model was simplified by fixing the same transmission range for all nodes and using the Boolean interference model. Initially, such a number of packets were placed to the network that the network was under heavy traffic. The forwarding methods to be compared were a deterministic MFR method, two randomized forwarding methods and opportunistic forwarding that includes an own slotted MAC scheme.

For each forwarding method, the dimensionless mean density of progress was evaluated as a function of transmission probability for different network densities. According to these results, opportunistic forwarding performs clearly better than other forwarding methods. Of the methods using the basic slotted ALOHA, randomized forwarding methods achieve approximately twice as

high performance as MFR. When the distribution of packets in the network was analyzed, it was noticed that when MFR is used, packets deterministically follow the same paths and the overall network utilization is low. Although randomized forwarding methods spread traffic more evenly to the network, there still exist bottleneck nodes that have significantly longer queue sizes than a typical node.

7.2 Further work

The problem of maximizing the mean density of progress in a large ad hoc network using simulations provides multiple potential directions for future research. Modifications to our simulation scenarios can be done by improving the local forwarding rules, modifying the slotted ALOHA MAC protocol or changing the underlying physical model.

As can be seen from Figures 6.13 and 6.14, even if randomized forwarding methods are used, there exist bottleneck nodes that have most of the packets in their queues and some areas of the network have very light traffic. As a further work, it could be studied whether it is possible to reach a better network utilization (and better $u(N_R, p)$) by taking into account also the queue sizes of neighboring nodes when making a local forwarding decision. The queue size information can be added to acknowledgment packets or to periodical location advertisements. For example, if the cost of forwarding a packet to a neighbor is proportional to the queue size at the neighbor and inversely proportional to the progress of the hop, a sending node forwards the packet to the neighbor with the smallest cost.

The MAC protocol could be modified to support transmission power control. Each node would then transmit with a sufficient power level to reach the next hop neighbor. This increases the spatial reuse in the network and accordingly the mean density of progress. Note that this kind of power control cannot be used with opportunistic forwarding because the next hop is selected only after data transmission. Another approach could be to avoid collisions using a distributed coordination among the nodes. If the network was partitioned with a predefined grid into small squares and each node knew the partitioning, it could be possible to allow only nodes in certain areas to content for the channel during a certain time slot. However, the size of a grid square and the scheduling among the squares need to be carefully selected.

The Boolean model for interference and path loss is overly simplified if compared to the real world. A potential open issue would be to use a more realistic interference model, such as the Physical Model presented in Section 3.2, and

find out if it had a significant effect on the results. Finally, node mobility could be added to the network model. However, even if a simple mobility model, such as the random waypoint model [50], is used, the computational effort can easily become infeasible because of the large number of nodes.

Bibliography

[1] Andrea J. Goldsmith and Stephen B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Communications*, 9(4):8 – 27, 2002.

[2] Robert E. Kahn. The organization of computer resources into a packet radio network. *IEEE Transactions on Communications*, 25(1):169 – 178, 1977.

[3] Norman Abramson. The ALOHA system — another alternative for computer communications. In *Proc. of AFIPS Fall Joint Computer Conference*, pages 281 – 285, Montvale, New Jersey, November 1970.

[4] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc of ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234 – 244, London, UK, August – September 1994.

[5] Sunil Kumar, Vineet S. Raghavan, and Jing Deng. Medium access control protocols for ad hoc wireless networks: a survey. *Ad Hoc Networks*, to appear.

[6] Raja Jurdak, Cristina Videira Lopes, and Pierre Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *IEEE Communications Surveys & Tutorials*, 6(1):2 – 16, 2004.

[7] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1 – 22, 2004.

[8] Mahesh K. Marina and Samir R. Das. Routing in mobile ad hoc networks. In *Ad Hoc Networks: Technologies and Protocols*. Springer, 2004.

[9] Vikas Kawadia and P. R. Kumar. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(1):76 – 88, 2005.

[10] Prasant Mohapatra, Jian Li, and Chao Gui. QoS in mobile ad hoc networks. *IEEE Wireless Communications*, 10(3):44 – 52, 2003.

[11] T. Bheemarjuna Reddy, I. Karthigeyan, B.S. Manoj, and C. Siva Ram Murthy. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad Hoc Networks*, 4(1):83 – 124, 2006.

[12] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in mobile ad hoc networks: Challenges and solutions. *IEEE Wireless Communications*, 11(1):38 – 47, 2004.

[13] Christian Bettstetter. On the connectivity of ad hoc networks. *The Computer Journal*, 47(4):432 – 447, 2004.

[14] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102 – 114, 2002.

[15] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28 – 36, 2002.

[16] Jean Carle and David Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *Computer*, 37(2):40 – 46, 2004.

[17] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88 – 97, Atlanta, Georgia, September 2002.

[18] Edoardo S. Biagioni and K. W. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *The International Journal of High Performance Computing Applications*, 16(3):112 – 121, 2002.

[19] Dragoş Niculescu. Communication paradigms for sensor networks. *IEEE Communications Magazine*, 43(3):116 – 122, 2005.

[20] Piyush Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[21] Christian Bettstetter and Christian Hartmann. Connectivity of wireless multihop networks in a shadow fading environment. In *Proc of ACM International Workshop on Modeling Analysis and Simulation of Wireless Mobile Systems*, pages 28 – 32, San Diego, California, September 2003.

[22] Lawrence G. Roberts. Aloha packet system with and without slots and capture. Technical Report ARPA Satellite System Note 8, Stanford Research Institute, Stanford, California, June 1972.

[23] Leonard Kleinrock and Fouad A. Tobagi. Packet switching in radio channels: Part I — carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400 – 1416, 1975.

[24] Fouad A. Tobagi and Leonard Kleinrock. Packet switching in radio channels: Part II—the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

[25] Phil Karn. MACA — a new channel access method for packet radio. In *Proc. of ARRL 9th Computer Networking Conference*, London, Canada, September 1990.

[26] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. *ACM SIGCOMM Computer Communication Review*, 24(4):212 – 225, 1994.

[27] Chane L. Fullmer and J.J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. *ACM SIGCOMM Computer Communication Review*, 25(4):262 – 273, 1995.

[28] IEEE 802.11 Working Group. *IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[29] William Stallings. *Wireless Communications & Networks*, chapter 14 Wi-Fi and the IEEE 802.11 Wireless LAN Standard, pages 421 – 462. Pearson Education International, 2nd edition, 2005.

[30] Shugong Xu and Tarek Saadawi. Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks. *IEEE Communications Magazine*, 39(6):130 – 137, 2001.

[31] Zygmunt J. Haas and Jing Deng. Dual busy tone multiple access (DBTMA) — a multiple access control scheme for ad hoc networks. *IEEE Transactions on Communications*, 50(6):975 – 985, 2002.

[32] Suresh Singh and C.S. Raghavendra. PAMAS — power aware multi-access protocol with signalling for ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 28(3):5 – 26, 1998.

[33] Jeffrey P. Monks, Vaduvur Bharghavan, and Wen mei W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proc. of IEEE INFOCOM*, pages 219 – 228, Anchorage, Alaska, April 2001.

[34] Eun-Sun Jung and Nitin H. Vaidya. A power control MAC protocol for ad hoc networks. In *Proc. of International Conference on Mobile Computing and Networking*, pages 36 – 47, Atlanta, Georgia, September 2002.

[35] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. In *Proc. of Wireless Communications and Networking Conference*, pages 1214 – 1219, Chicago, Illinois, September 2000.

[36] Young-Bae Ko, Vinaychandra Shankarkumar, and Nitin H. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *Proc. of IEEE INFOCOM*, pages 13 – 21, Tel-Aviv, Israel, March 2000.

[37] Yu Wang and J.J. Garcia-Luna-Aceves. Spatial reuse and collision avoidance in ad hoc networks with directional antennas. In *Proc. of IEEE Global Telecommunications Conference*, pages 112 – 116, Taipei, Taiwan, November 2002.

[38] Asis Nasipuri, Jun Zhuang, and Samir R. Das. A multichannel CSMA MAC protocol for multihop wireless networks. In *Proc. of Wireless Communications and Networking Conference*, pages 1402 – 1406, New Orleans, Lousiana, September 1999.

[39] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proc. of International Symposium on Parallel Architectures, Algorithms and Networks*, pages 232 – 237, Dallas, Texas, December 2000.

[40] Shih-Lin Wu, Yu-Chee Tseng, Chih-Yu Lin, and Jang-Ping Sheu. A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. *The Computer Journal*, 45(1):101 – 110, 2002.

[41] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493 – 506, 2004.

[42] Peng Lin, Chunming Qiao, and Xin Wang. Medium access control with a dynamic duty cycle for sensor networks. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 1534 – 1539, Atlanta, Georgia, March 2004.

[43] Carlos de Morais Cordeiro, Hrishikesh Gossain, and Dharma P. Agrawal. Multicast over wireless mobile ad hoc networks: Present and future directions. *IEEE Network*, 17(1):52 – 59, 2003.

[44] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. In *Proc. of IEEE INFOCOM*, volume 3, pages 1707 – 1716, New York, NY, June 2002.

[45] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proc. of Third ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 194 – 205, Lausanne, Switzerland, June 2002.

[46] Dimitri P. Bertsekas and Robert G. Gallager. *Data Networks*, chapter Distributed Asynchronous Bellman-Ford Algorithm, pages 325 – 333. Prentice Hall, 1987.

[47] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[48] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proc. of IEEE International Conference on Communications*, pages 70 – 74, New Orleans, Lousiana, June 2000.

[49] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proc. of IEEE International Multi Topic Conference INMIC*, pages 62 – 68, Lahore, Pakistan, December 2001.

[50] David B. Johnson and David A. Malz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Henry F. Korth, editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1996.

[51] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pages 90 – 100, New Orleans, Lousiana, February 1999.

[52] Zygmunt J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of IEEE International Conference on Universal Personal Communications*, pages 562 – 566, San Diego, California, October 1997.

[53] Frederick Ducatelle, Gianni Di Caro, and Luca Maria Gambardella. Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. *International Journal of Computational Intelligence and Applications*, 5(2):169 – 184, 2005.

[54] Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137 – 172, 1999.

[55] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[56] Ivan Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.

[57] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proc. of Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999.

[58] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609 – 616, 2001.

[59] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[60] Brad Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of International Conference on Mobile Computing and Networking*, pages 243 – 254, Boston, Massachusetts, August 2000.

[61] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proc. of International Conference on Mobile Computing and Networking*, pages 267 – 278, Annapolis, Maryland, June 2003.

[62] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proc. of Annual Symposium on Principles of Distributed Computing*, pages 63 – 72, Boston, Massachusetts, July 2003.

[63] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. In *Proc. of ACM Second Workshop on Hot Topics in Networks*, Cambridge, Massachusetts, November 2003.

[64] Sanjit Biswas and Robert Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proc. of Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pages 133 – 144, Philadelphia, Pennsylvania, August 2005.

[65] Marc Heissenbüttel, Torsten Braun, Thomas Bernoulli, and Markus Wälchli. BLR: Beacon-less routing algorithm for mobile ad hoc networks. *Computer Communications*, 27(11):1043 – 1126, 2004.

[66] Saad Biaz and Yiming Ji. A survey and comparison on localisation algorithms for wireless ad hoc networks. *International Journal of Mobile Communications*, 3(4):374 – 410, 2005.

[67] Saumitra M. Das, Himabindu Pucha, and Y. Charlie Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *Proc. of IEEE INFOCOM*, pages 1228 – 1239, Miami, Florida, March 2005.

[68] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proc. of International Conference on Mobile Computing and Networking*, pages 76 – 84, Dallas, Texas, October 1998.

[69] Ivan Stojmenovic and Pedro Eduardo Villaneuva Peña. A scalable quorum based location update scheme for routing in ad hoc wireless networks. Technical report, University of Ottawa, September 1999.

[70] Seung-Chul M. Woo and Suresh Singh. Scalable routing protocol for ad hoc networks. *Wireless Networks*, 7(5):513 – 529, 2001.

[71] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proc. of International Conference on Mobile Computing and Networking*, pages 120 – 130, Boston, Massachusetts, August 2000.

[72] Christine T. Cheng, Howard L. Lemberg, Sumesh J. Philip, Eric van den Berg, and Tao Zhang. SLALoM: A scalable location management scheme for large mobile ad-hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 574 – 578, Orlando, Florida, March 2002.

[73] Wolfgang Kieß, Holger Füßler, Jörg Widmer, and Martin Mauve. Hierarchical location service for mobile ad-hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(4):47 – 58, 2004.

[74] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325 – 349, 2005.

[75] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000.

[76] Stephanie Lindsey and Cauligi S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Proc. of IEEE Aerospace Conference*, pages 1125 – 1130, Big Sky, Montana, March 2002.

[77] Mohamed Younis and Tamer Nadeem. Energy efficient MAC protocols for wireless sensor networks. In Ahmed Safwat, editor, *Wireless Ad-Hoc and Sensor Networks*. Kluwer Academic Publishers, to appear.

[78] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1):2 – 16, 2003.

[79] Joanna Kulik, Wendi Heinzelman, and Hari Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2/3):169 – 185, 2002.

[80] Li Li and Joseph Y. Halpern. A minimum-energy path-preserving topology-control algorithm. *IEEE Transactions on Wireless Communications*, 3(3):910 – 921, 2004.

[81] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333 – 1344, 1999.

[82] Esa Hyytiä and Jorma Virtamo. On load balancing in a dense wireless multihop network. submitted for publication, 2005.

[83] Dragoş Niculescu and Badri Nath. Trajectory based forwarding and its applications. In *Proc. of International Conference on Mobile Computing and Networking*, pages 260 – 272, San Diego, California, September 2003.

[84] François Baccelli, Bartłomiej Błaszczyszyn, and Paul Mühlethaler. An Aloha protocol for multihop mobile wireless networks. In *Proc. of ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp, Belgium, August–September 2004.

[85] Henri Koskinen and Jorma Virtamo. Probability of successful transmission in a random slotted-aloha wireless multihop network employing constant transmission power. In *Proc. of International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 191 – 199, Montreal, Canada, October 2005.