HELSINKI UNIVERSITY OF TECHNOLOGY

Jarno Nousiainen

Forwarding Capacity of an Infinite Homogeneous Wireless Network

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 March 2008

Author: Jarno Nousiainen

Title: Forwarding Capacity of an Infinite Homogeneous Wireless Network

Date: 1st March 2008 **Number of pages:** 66+6

Chair: S-38 Teletraffic Theory

Supervisor: Professor Jorma Virtamo

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

An ad hoc network is a wireless network independent of any fixed infrastructure where the nodes communicate with each other in a multihop fashion. In the absence of centralized control, the nodes are responsible for all network activity, which includes discovering the route to the destination and forwarding packets towards it. We begin this thesis with a short introduction to ad hoc networks and the factors affecting their performance, namely medium access control (MAC) and routing. We also consider wireless sensor networks (WSN), a special case of ad hoc networks, that offer a wide range of proposed applications for large ad hoc networks.

When the ad hoc network is very large, the macroscopic level, corresponding to the scale of an end-to-end path, and the microscopic level, corresponding to the scale of a single hop, can be separated. The macroscopic level routing protocol, treating the network as a continuous medium, provides the direction of packet flow to the microscopic level where the packets are forwarded based on this information according to the rules of the microscopic level forwarding method. Considering one direction at a time, there exists a certain maximum flow of packets that can be supported. Generally, this maximal sustainable directed packet flow depends on the network properties and the used medium access control (MAC) protocol. The capacity of the network can be divided between different direction, e.g., via time sharing.

In the main contribution of the thesis, we model a large ad hoc network and devise a simulation algorithm for obtaining an upper bound for the maximal forwarding capacity under that model. The Moving window algorithm (MWA) that is based on an augmentation of the max-flow min-cut theorem is then improved to produce tighter upper bounds for the maximal capacity. The results are compared to the capacities of existing forwarding methods, providing feasible lower bounds, and the optimal capacities of networks with regular structure. The tightest obtained upper bound is about three times the maximum performance achieved with existing forwarding methods.

Keywords: wireless multihop networks, routing, medium access control, forwarding capacity, density of progress, graph algorithms, flow networks

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Jarno Nousiainen		
Otsikko:	Äärettömän homogeenisen langattoman verkon välityskapasiteetti		
Päivämäärä:	1. maaliskuuta 2008	**Sivumäärä:**	66+6
Professuuri:	S-38 Teleliikenneteoria		
Työn valvoja:	Professori Jorma Virtamo		
Työn ohjaaja:	TkT Pasi Lassila		

Ad hoc -verkko on langaton verkko, joka toimii ilman kiinteää verkkoinfrastruktuuria ja jossa päätelaitteet voivat viestiä keskenään toistensa välityksellä. Keskitetyn valvonnan puuttuessa verkon solmut ovat itse vastuussa kaikesta verkon toiminnallisuudesta, joka pitää sisällään niin reitin löytämisen kohteeseen kuin pakettien välittämisen sitä kohti. Työ alkaa lyhyellä ad hoc -verkkojen ja niiden suorituskykyyn vaikuttavien tekijöiden, pääsynvalvonnan (MAC) ja reitityksen, esittelyllä. Lisäksi käsitellään langattomia sensoriverkkoja, jotka muodostavat ad hoc -verkkojen erikoistapauksen ja joilla on lukuisia ehdotettuja sovelluksia laajoissa ad hoc verkoissa.

Kun ad hoc -verkko on laaja, voidaan siinä erottaa makroskooppinen taso, joka vastaa päästä päähän -polun mittakaavaa, ja mikroskooppinen taso, joka vastaa yksittäisen hypyn mittakaavaa. Makroskooppisen tason reititysprotokolla näkee verkon jatkuvana väliaineena, jossa kulkevan pakettivuon suunnan se välittää mikroskooppiselle tasolle. Mikroskooppisen tason välitysmenetelmä puolestaan perustaa päätöksensä tälle informaatiolle. Yhtä suuntaa kerrallaan tarkasteltaessa on olemassa suurin pakettivuo, joka on mahdollista saavuttaa. Tämä maksimaalinen suunnattu pakettivuo riippuu verkon ominaisuuksista ja voidaan jakaa esimerkiksi aikaperusteisesti eri suuntien välillä.

Työn keskeisessä osassa mallinnetaan laaja ad hoc -verkko ja laaditaan simulointialgoritmi suurimman ylläpidettävän vuon ylärajan löytämiseksi. Liukuvan ikkunan algoritmi (MWA) perustuu laajennettuun max-flow min-cut -lauseeseen. Algoritmia kehitetään edelleen yhä tiukempien ylärajojen tuottamiseksi. Algoritmilla saatuja tuloksia verrataan olemassa olevien välitysmenetelmien saavuttamiin kapasiteetteihin, jotka ovat alarajoihin suurimmalle mahdolliselle välityskapasiteetille, sekä säännöllisten verkkojen suorituskykyyn. Tiukin löydetty yläraja on noin kolminkertainen suurimpaan olemassa olevilla välitysmenetelmillä saavutettuun välityskapasiteettiin verrattuna.

Avainsanat: langattomat monihyppyverkot, reititys, MAC, välityskapasiteetti, etenemisen tiheys, graafialgoritmit, virtausverkot

Acknowledgements

This thesis was written in the Networking Laboratory of Helsinki University of Technology, now known as the Department of Communications and Networking, as a part of the ABI project.

I would like to express my gratitude to my supervisor, Professor Jorma Virtamo, for his valuable guidance and ideas, and my instructor Pasi Lassila for his ideas, comments, and tireless guidance. I would also like to thank all the people who helped me during the work.

Finally, I would like to thank my family and friends.

Espoo, 1st March 2008

Jarno Nousiainen

Contents

Chapter 1

Introduction

1.1 Background

An ad hoc network is a decentralized network that does not rely on any pre-existing infrastructure. It allows a rapid deployment of wireless nodes that act the dual role of both terminals and routers.

The idea of a wireless self-configuring randomly deployed multi-hop network is not a new one, and the roots of ad hoc networking can be traced back to the ALOHA project initiated in the late 1960s [1]. Though the ALOHA protocol itself was a single-hop protocol, it created the basis for the development of ad hoc networking by introducing a suitable distributed channel-access method.

The work on the first multi-hop wireless network, The DARPA Packet radio network (PRNET), began in the early 1970s [2]. Since the PRNET, ad hoc networks have been the suggested solution for many military communication applications, but in the consumer segment they never really gained much attention at the time.

When developing a standard for wireless local area networks (WLAN), the Institute of Electrical and Electronic Engineering (IEEE) replaced the term packet-radio network with the current term ad hoc network. Only with the appearance of inexpensive WLAN solutions during the 1990s have ad hoc networks become such a popular research topic.

The commercial applications still hardly satisfy the criteria of pure ad hoc networking, but the recent advances in the areas of micro electronics have made possible a new form of ad hoc networking – the wireless sensor networks (WSN). Although having many differences to the more traditional ad hoc networks, the WSNs have several features that make them particularly interesting in terms of ad hoc networking. One such feature is the very high number nodes in many of the proposed applications.

1.2 Problem statement

In a large, dense wireless multihop network, a typical distance between a randomly selected source-destination pair is much greater than the distance between two adjacent nodes that are able to communicate with each other directly. Thus, an average path in the network consists of several hops while the nodes along the path act as relays.

We separate the macroscopic level, corresponding to the distance between source-destination pairs, and microscopic level, corresponding to the distance between adjacent nodes [3]. The macroscopic level sees the network as a fabric forming a homogenous, continuous medium, and the routes are, in general, smooth geometric curves. At the microscopic level, we consider a single node and its immediate neighbors, giving significance only to the direction in which a certain packet is forwarded.

We focus on the latter, that is, the microscopic level. On the microscopic level, given a progress metric, there exists a maximum flow of packets – total progress of packets per unit area per unit time, i.e., density of progress (3.2) – that the network can sustain in a given direction. This maximal capacity can be divided between flows traversing in different directions by, e.g., a simple time sharing mechanism. Generally, the maximum packet flow that the network can support depends on the MAC protocol providing the access to the wireless channel, but we, in particular, are interested in finding an upper bound for the maximal achievable forwarding capacity in the network specified in Chapter 3.

The approach is to model the network as a random graph. We review the maximal flows in traditional flow networks and derive a theoretical expression for the upper bound of the forwarding capacity in the wireless equivalent. Because of the overwhelming size of the problem, simulation methods are proposed for obtaining a numerical value for the maximal forwarding capacity. These values are compared to the ones of viable forwarding schemes (lower bounds for the searched value) and to the maximal capacities of networks corresponding to regular lattices.

1.3 Structure of the thesis

Chapter 2 gives a brief overview on wireless multihop networks, namely ad hoc and sensor networks. The introductory sections give definitions for both network types, as well as, a few proposed applications. The challenges of MAC and routing are covered separately for both.

The network model used in this study is presented in Chapter 3. After listing the assumptions and defining the performance measure, we model the wireless network as a graph and present the related concepts of connectivity and percolation.

In Chapter 4, we introduce the well-known max-flow min-cut theorem and examine its significance to wireless networks. Finally, the chapter is concluded with a study of the maximal achievable flow in regular networks.

Chapter 5 presents an algorithm for simulating the maximal forwarding capacity over a single timeslot. The algorithm is devised and three different versions are presented to tighten the upper bound of the capacity. For additional intuition and to obtain a lower bound for the upper bound, approximative methods are finally considered.

Chapter 6 represents simulation results from actual forwarding methods that give a certain lower bound for the achievable maximum flow.

Chapters 7 and 8 give a summary of the results and conclude the study.

Chapter 2

Wireless multihop networks

2.1 Ad hoc networks

The earliest applications of packet radio networks, as they were called back then, were mainly for military purposes and have been studied since the 1970s. The appearance of inexpensive WLAN solutions during the 1990s made ad hoc networks a popular research topic, and the increasing availability of wireless devices ever since has made ad hoc networking one of today's most active fields in communications research.

The term ad hoc network itself refers to a computer network with no fixed infrastructure where the nodes usually communicate in a wireless fashion. The decentralization of the network means that nodes are responsible for all network activity, which includes discovering the route to the destination and forwarding packets towards it. Since topology changes due to node mobility are also possible, the connections can only be established for the duration of the communication session. An ad hoc network (see Figure 2.1) does not rely on pre-existing infrastructure, which makes one of the most attractive features of ad hoc networking – the random deployment of the nodes – possible. It should also be possible for the nodes to dynamically join and leave with minimal disturbance to the network.

The various uses of wireless ad hoc networks include, in addition to the initial military applications, communication in areas without adequate wireless coverage like rescue missions in remote tracts or communication in areas where the infrastructure has been destroyed due to, for example, a natural disaster or a war. The rapid deployment of an ad hoc network may also be the most appropriate solution in situations like law enforcement operations or exhibitions and conferences. A common factor for all of the above applications on some level is the collaboration of the whole network towards a common goal. Thus, they avoid a question that would play a large role in fully commercial applications: why would somebody use his limited resources to forward someone else's traffic? One yet challenging solution could be billing, but

so far commercial uses mostly include connecting portable machines like PDAs to existing networks in an ad hoc fashion.

The performance of an ad hoc network can be measured using indicators such as throughput, latency, energy consumption, and fairness, and it is closely related to the concepts of routing and medium access control (MAC). Routing is responsible for providing the paths for the traffic, while MAC provides addressing and channel access control mechanisms. For the performance to be as good as possible, routing and MAC have to be usually designed to work together.

2.1.1 Medium access control

The frequency spectrum is highly controlled, and the applications have thus only limited channel bandwidth. This means that the available capacity should be used as efficiently as possible. What makes this difficult is that the wireless medium is prone to errors and medium specific problems such as the hidden and exposed terminal problems, signal fading, noise, and interference. The Medium Access Control (MAC) protocol is responsible for providing the nodes with access to the medium. The coordination of the access from the active nodes should be fair so that the use of the scarce resource is expedient. Various types of MAC schemes have been developed for different types of ad hoc networks.

The first division made among MAC schemes is usually between the contention-free and contention-based schemes (although the naming may vary). The first group includes controlled channel access techniques that assign the nodes to different time slots (TDMA), frequency bands (FDMA), or other data channels (e.g., CDMA). Because the channel is primarily allocated to a single pair of nodes, there is no contention for the channel. This works well if the traffic load is high, but with less

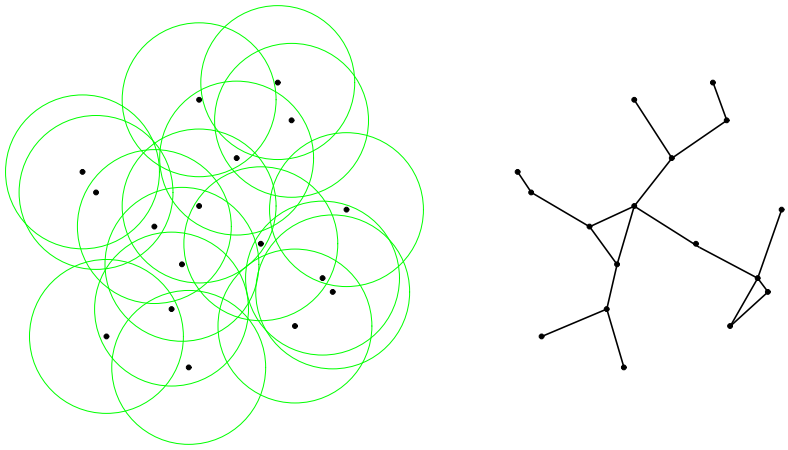

Figure 2.1: An example of an ad hoc network – ad hoc nodes with their transmission radii shown and the resulting network graph.

traffic, all the channels are not utilized, and the delays become unnecessarily high.

Because contention-free schemes are difficult to implement without a static network using centralized control, contention-based techniques are more applicable to ad hoc networks. The contention-based schemes include random access and dynamic reservation protocols. In random access methods, like ALOHA [1], a node with data to transmit may access the channel with a certain probability. This corresponds to a distributed scheduling algorithm that randomly allocates the channel to requesting nodes, and evidently leads to collisions. The probability of a collision can be reduced by using time slots (slotted ALOHA [4]) or listening to the channel before sending (CSMA [5]).

To completely avoid the interference between nodes using a shared channel, dynamic reservation/collision resolution protocols use control packets to reserve the channel (e.g., MACA [6]), or combine control packets with carrier sensing (e.g., FAMA [7]). These methods cope better with hidden and exposed terminal problems, but require more complex nodes, and the control traffic reduces the bandwidth available for data traffic.

Furthermore, the MAC schemes may be classified as sender- vs. receiver-initiated, single- vs. multiple-channel, power aware, directional antenna based etc. For a survey on MAC protocols for ad hoc networks and a classification see, e.g., [8] or [9].

2.1.2 Routing

For communication within the network to be possible, a routing protocol is required to establish a connection between the participating nodes. Because an ad hoc network does not have a fixed infrastructure or centralized control, the nodes are responsible for performing the routing functions themselves.

The special nature of large ad hoc networks places some requirements on the routing protocol. Movement of the nodes causes changes to the network topology and the routing protocols need to be able to adapt to these changes. At the same time the routing overhead should be kept minimal since the bandwidth in the shared wireless channel is limited. Efficient use of the channel is crucial also to save battery power, which is an issue with mobile devices.

Traditional ad hoc routing protocols fall into two general categories: proactive (table driven) and reactive (on-demand). Proactive routing protocols maintain information about the whole network in every single node. With a complete picture of the network, determining a route is fast, but whenever the topology changes, all the routing tables need to be updated. This means that recurrent changes in the topology, especially in case of a large network, cause the amount of overhead traffic to increase significantly. Hence, proactive routing protocols, including Destination-Sequenced Distance Vector (DSDV) [10], Fish-eye State Routing (FSR) [11], and Optimized

Link State Routing (OLSR) [12], perform best when the nodes have low mobility compared to the frequency with which they transmit data.

Reactive routing protocols do not maintain routing tables about the whole network. Instead, a route is only found when there is data to send. This reduces the amount of routing traffic caused by the changes in the network topology and also the storage capacity needed. Whenever the information about the required route is not available, a node starts a route discovery procedure, causing a significant delay before the packet can be transmitted. Thus, reactive ad hoc routing protocols, including Dynamic Source Routing (DSR) [13], Ad Hoc On-Demand Distance Vector (AODV) [14], and Temporally Ordered Routing Algorithm (TORA) [15], are most useful when the network topology changes constantly or when data transmissions are infrequent and delay tolerant.

There exist also hybrid protocols that combine both proactive and reactive routing protocols. Since proactive and reactive routing schemes work well in opposite types of networks, it is possible to utilize them hierarchically to increase the performance compared to the pure proactive and reactive protocols. Examples of hybrid protocols include Zone Routing Protocol (ZRP) [16] and AntHocNet [17].

Traditional routing protocols presented above collect and store information about the network topology, and it is questionable whether this kind of approach is feasible when the number of nodes reaches hundreds or thousands. Geographic routing protocols (see [18] for an overview) are a promising alternative for traditional methods in large ad hoc networks, and they use the geographic locations of the nodes as a base for their routing decisions. If the location of the destination is known, a node needs only local information about its own and its neighbors' locations to be able to forward the packet. Hence, the scalability of such protocols is mostly dependent on the location service (see [19] or [20] for comparison) which performs the tracking of the destination nodes.

The most obvious way of making the decision about the next hop is to try to forward the packet as far as possible with respect to a given progress metric. These greedy forwarding methods include, for example, Most Forward within Radius (MFR) [21] and Geographical Distance Routing (GEDIR) [22]. However, to work properly a geographic routing protocol needs to be able to handle routing around concave nodes, i.e., nodes that have no neighbors in the direction of the destination (forward neighbors). Typically routing algorithms that guarantee packet delivery work as follows: greedy forwarding is used as long as possible, but when packet reaches a dead end, a recovery procedure such as face routing [23] is taken into practice. Geographic routing protocols using face routing include Greedy Face Greedy (GFG) [23], Greedy Other Adaptive Face Routing (GOAFR) [24], and Greedy Perimeter Stateless Routing (GPSR) [25].

2.2 Sensor networks

Only the recent advances in the areas of wireless network technologies and small-scale electronics have made it possible to develop low-cost sensor nodes useful for building wireless sensor networks with numerous intended applications. The following section gives a brief overview of these sensor networks. For more information, see for example [26, 27].

A wireless sensor network (WSN) consists of a large number of densely deployed nodes that are used for sensing a phenomenon such as, for example, temperature or motion. The application may require random deployment, which means that the used protocols and algorithms have to be self-organizing. Such an application could be, e.g., disaster area monitoring where the studied region is inaccessible or rough, and the nodes need to be, for example, dropped on site from a helicopter.

The idea of a sensor network is to deploy a sensor field consisting of large number of nodes either inside or near the investigated phenomenon. The nodes play the dual role of data originator and router as they forward the required data from the sensing area towards the sink. The sink (gateway node) connects the sensor network to the task manager (end user) through, e.g., Internet or satellites. Figure 2.2 shows an example of the structure of a WSN.

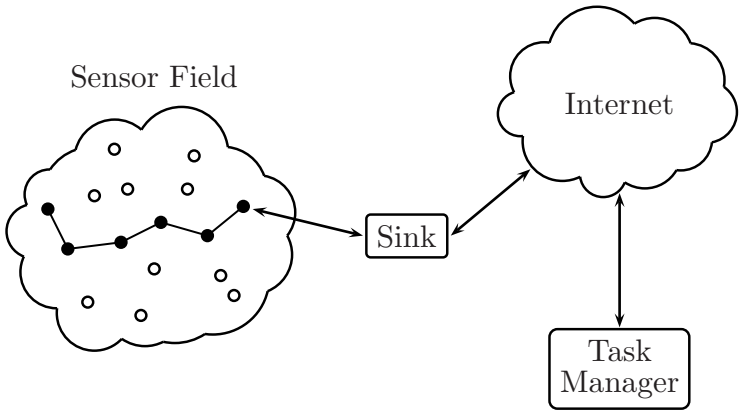

Figure 2.2: The structure of a sensor network.

The earliest applications for WSNs were strongly motivated by military needs. WSNs can be used for example for battlefield surveillance, chemical attack detection or target acquisition. Sensor networks have also been suggested for intrusion detection to replace land mines. The most typical civil application for a WSN is environment and habitat monitoring [28, 29]. The networks can be used for tracking animals, observing and forecasting weather, detecting and monitoring pollution, floods, and forest fires etc. [30, 31]. Other civilian uses include health applications such as patient monitoring and diagnostics, as well as, construction of smart structures and buildings [32, 33].

2.2.1 Sensor node

A sensor node is a small autonomous device comparable to a simple computer. It usually consists of a few main components that are used for sensing, processing, and communication purposes (see Figure 2.3). The sensing unit consists of at least one sensor and an analog-to-digital converter. The data coming from the sensing unit goes to the processing unit, which is responsible for processing the data and transmitting only the data required for carrying out the assigned tasks. The processing unit also handles the cooperation between the nodes. The communication is typically handled with a radio transceiver, but optical and infrared solutions also exist. The last main component in addition to the previous units is the power unit, usually a battery. The nodes may also have some additional components such as a location finding unit, an energy harvesting unit or a mobilizer depending on the application.

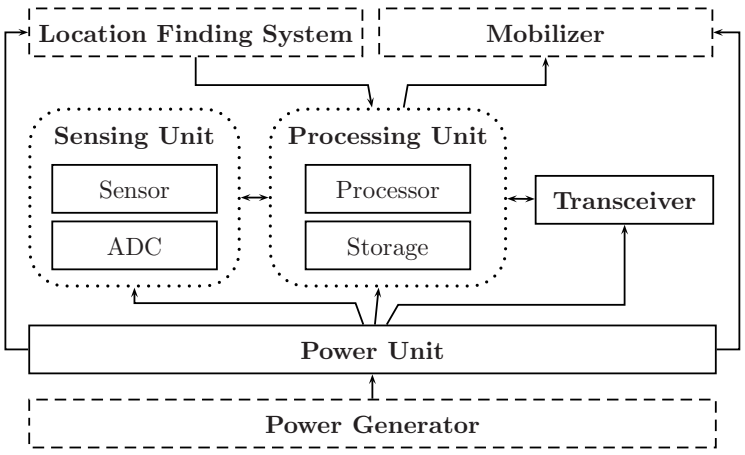

Figure 2.3: The components of a sensor node.

Typically, the nodes need to be small in size. For example, some future applications require the size of a single node to be as small as under one cubic centimeter. The cost of such a node should also be as small as possible, because the network consists of a large number of these nodes. The cost of the whole network has to be lower than it would be to use wired sensors or other substitutive method in order to justify the use of the technology. It is also important to keep the production costs low, because many applications require the nodes to be dispensable. It is not possible to rescue the nodes from a forest fire or volcano, or gather them back from a disaster area. Besides, the low production cost and the exposure to harsh weather conditions and physical stress make the nodes prone to failures.

2.2.2 Medium access control

Though a wireless sensor network is a specialized version of an ad hoc network, there are some conceptual differences between a WSN and a traditional ad hoc network

causing fundamental adjustments to the protocol design.

Typically, the number of nodes in a WSN can be several orders of magnitude higher than the number of nodes in an ad hoc network. Also the mean number of neighbors per node is usually higher in sensor networks. This requires scalable methods for large, densely deployed networks as both forming the basic infrastructure and efficiently sharing the communication resources between the nodes become more challenging. The high density of nodes increases the collision probability. Because sensor networks often have lower data rate requirements and higher delay tolerance, the energy used for retransmission is usually a greater loss than the time spent waiting for a better transmission spot. From the MAC viewpoint, this also means that the signaling overhead should be minimized to prevent further collisions.

This brings us to the biggest issue in sensor network MAC design — the amount of energy available. Since the batteries have to be small, and it might be impossible to recharge or replace them, the sensor node lifetime is strongly dependent on the battery lifetime. From the three tasks of the node, sensing, data processing, and communication, the last one is, without a doubt, the most critical energywise [34, 35]. According to [34], it may take even more than 100 000 times less energy to execute a 32-bit instruction than to send 100 bits for 100 m.

The number of energy-consuming transmissions and receptions can be limited by limiting collisions, overhearing, and overhead. Still, one critical transceiver operation remains. Idle listening consumes nearly as much energy as the active use of the receiver (transmitting requires somewhat more energy than receiving). The energy wasted while listening to an empty channel – excluding possible carrier sensing, which is useful – can account for a significant amount of the total energy used [36].

As a solution to the ensuing lifetime problem, the MAC protocols typically turn the receiver off for certain sleep periods. These sleep periods bring up several new issues. Frequent changes between active and sleep states may end up consuming more energy than keeping the node active the whole time. Moreover, the sleep cycle of the transmitting and receiving node has to be the same. This requires clock synchronization, whose accuracy is limited by the used clock crystals that tend to be cheap and thus inaccurate in order to keep the cost of the node small. For more on clock synchronization, see [37].

The attempts to keep the cost of a single node as small as possible, as well as the energy limitations, also result in limited processing and memory capacities. This means that the conventional layered architecture may be too heavy, and complex algorithms cannot be implemented. Additionally, the on-off periods, node failures, and possible mobility make topology changes much more frequent in sensor networks.

Various MAC protocols have been proposed in the literature for wireless sensor networks (see [38] for a survey). The most common classification is still made between the contention-free and contention-based protocols. The first group contains sched-

uled (TDMA-based) protocols that reserve the channel to a certain node at a time. The common schedule reduces energy consumption and limits collisions, idle listening, and overhearing. Creating and maintaining the schedule requires additional messages though, and topology changes make the schedule maintenance even more complex. A new node entering the network must wait until it learns the schedule or separately join it before it can utilize the channel. Also nodes with synchronization shifts cause problems. The Traffic-Adaptive Medium Access protocol (TRAMA) [39] represents an example of a scheduled MAC protocol that tries to lose some of the disadvantages of a reservation-based protocol by offering random access periods for signaling and scheduled access periods for contention free data exchange.

The latter group of contention-based protocols contains the unscheduled (random, CSMA-based) methods that have the advantage of simplicity. The nodes do not have to maintain or share state information, which leads to less messages and smaller memory requirements. The network adapts better to changes in the traffic conditions or topology, because the resources can be reserved on a demand basis, and new nodes do not have to wait until they have obtained the schedule. On the downside are the increased number of collisions and increased idle listening. An example of a contention-based protocol is the spatial Correlation-based Collaborative Medium Access Control protocol (CC-MAC) [40], which exploits the spatial correlation between sensor nodes subject to the observed event to determine which nodes transmit their data, since it might not be necessary for all the nodes to transmit.

Though there are several proposed protocols, none of them is accepted as the final or adequate solution to the various challenges related to the field, and there is still a lot of research to be done on the area of sensor network MAC protocols. The existence of a general MAC protocol flexible enough to support various applications still remains an open question.

2.2.3 Routing

In addition to the characteristics of a WSN affecting the general protocol design and the cross-layer interaction, i.e., the large number of nodes, limited energy, and changing topology, there are some features that are directly related to routing.

Firstly, a global addressing scheme for the nodes is not possible due to the amount of overhead it would cause, because of the large number of nodes. Since the nodes do not have a global identification, classical IP-based protocols are not applicable.

Unlike in typical ad hoc networks, the traffic in a sensor network is usually content based. This means that receiving the data is more important than knowing the exact node that sent it. In a sensor network multiple sensors work for the same goal, and instead of the point-to-point traffic of many other networks, the traffic is usually many-to-one as the nodes propagate the sensed data towards the sink or one-to-many

as the sink sends queries to the nodes.

The sensed data usually has significant redundancy since multiple nodes may generate similar data in the area around the phenomenon. By aggregating such data in intermediate nodes, it is possible to improve energy and bandwidth utilization, which is of critical importance. In data aggregation the data from different sources is combined to reduce the number of packets sent. Such reduction can be made by removing duplicates (suppression), finding the minimum/maximum or calculating the average from the data [41].

Once again, much attention has been given to developing sensor network routing protocols since the requirements for such may differ depending on the application and network architecture. According to [42] most of the various existing sensor network routing protocols can be classified into three main categories. These categories are data-centric, hierarchical, and location-based (geographic) protocols.

Due to the lack of global ID, addressing a specific node or a specific set of nodes is difficult. To overcome this challenge, data-centric routing protocols gather and route data based on the properties of the data. This means that instead of being interested in a certain node, the user wants to know about an attribute of the phenomenon. Two basic realizations exist: In Sensor Protocol for Information via Negotiation (SPIN) [43], sensors advertise new data by broadcasting ADV-packets. The sink has the possibility to query that data with request (REQ) packets. The data is sent to the nodes that requested it by DATA packets. The Direct Diffusion protocol [44] works the other way around, and the sink queries sensors for certain data by flooding interests. Many data-centric protocols based on these two, as well as others, have also been proposed.

Scalability is one of the key issues when a routing protocol for sensor networks is considered. Hierarchical protocols group sensor nodes into clusters, from which a cluster-head is chosen. The cluster-heads may form yet another level of clusters depending on the size of the network. Besides scalability, hierarchical protocols offer a way to conserve energy. A cluster-head aggregates data from its cluster and sends the aggregated data to the sink on behalf of the other nodes in the cluster. Though energy is saved as a whole, the practice drains resources from the cluster-head. Some protocols assume the use of specialized cluster-head nodes that are less limited or that the cluster-heads have a direct connection to the sink. In homogeneous networks, the cluster-heads need to be changed. In the Low-Energy Adaptive Clustering Hierarchy protocol (LEACH) [45] nodes become cluster heads with a certain probability for a predetermined period of time.

Location-based routing protocols for sensor networks are still few in number. The information about node locations can be utilized to calculate minimum energy paths from a node to the sink, and to disseminate a query only to a particular area. The possibility to target a query to a specific region can reduce the number of trans-

missions significantly when only that area is of interest. Some geographic routing protocols designed for mobile ad hoc networks are also suitable for sensor networks with less or no mobility. It is required, though, that the protocol is energy aware. Examples of location-based protocols include MECN [46] and PRADA [47].

Chapter 3

Network model

3.1 Assumptions

The network consists of static nodes that communicate with each other over a wireless medium. Each node has an omnidirectional antenna, i.e., signals can be received from and transmitted to all directions. The transmission power is the same for all the nodes, resulting in a common fixed transmission radius R [m]. All the communication takes place in the same frequency, and a node is able to hear all the transmissions from nodes within its communication range, but none from those outside. Simultaneous transmissions interfere with each other in a way to be determined later, which may cause collisions and loss of data. The nodes are assumed to be reliable, so node failures are not considered.

The studied networks are assumed to be very large or even infinite. This allows us to delimit the type of traffic considered in a very propitious way. When the overall number of nodes in the network is large, two randomly selected nodes are, on the average, much further apart from each other than two neighboring ones. Thus, if the nodes communicating with each other are assumed to be random, a route between a source and a destination typically consists of a large number of hops. Therefore the amount of relay traffic in a specific area of the network is much higher than the amount of traffic that originates from or terminates to the area. This allows us to concentrate purely on the relay traffic and omit the originating and terminating traffic from the model. No traffic matrix or distribution is needed, but we simply study the amount of traffic that can be relayed through the network.

The system is assumed to be synchronous, and time is assumed to be divided into slots. The different directions are treated independently, and the progress of a packet forwarded during a time slot is calculated to a direction specified by the time slot. Thus, the problem considers maximizing the flow of packets in a given direction and different directions are handled, for example, using time sharing. The packet

size is fixed for the transmission time to match the length of a time slot. As the transmissions can only start at the beginning of a time slot, packets overlap either completely or not at all. If successful transmissions need to be acknowledged, the size of such an acknowledgement packet is assumed to be very small compared to the size of a data packet. The time required for the acknowledgement is thus assumed to be included in the length of the time slot. Generally, when studying upper bounds, the issue is not essential since collisions do not occur in optimal schedules.

Apart from the regular grid networks also studied in this work, the nodes are assumed to be located according to the spatial Poisson point process in two dimensions. The intensity of the process, referred to as the node density, is denoted by λ [$1/\mathrm{m}^2$]. Since the transmission radius R is common for all the nodes, the density of the network can now be described with the average number of nodes within the transmission range of a node, $N_R = \lambda\pi R^2$, which is dimensionless.

3.2 Mean density of progress

The performance of the network from a single node's point of view can be defined as the average progress of a packet in a given direction per timeslot [21]. This mean progress, D [m], is given by

$$\begin{aligned} D \;=\; & \mathbb{P}(\text{node transmits}) \cdot \mathbb{P}(\text{no collisions} \mid \text{node transmits}) \\ & \cdot \mathbb{E}[\text{progress of a packet} \mid \text{successful transmission}]. \end{aligned} \tag{3.1}$$

While the mean number of nodes in a differential area element equals $\lambda \cdot dA$, a network level measure for the performance, the total progress of packets per unit time per unit area or the mean density of progress, I [$1/(\mathrm{m} \cdot \mathrm{s})$], can be expressed by means of D as

$$I = \frac{\lambda \cdot dA \cdot D}{dA \cdot \Delta t} = \frac{\sqrt{\lambda}}{\Delta t} \cdot u, \tag{3.2}$$

where Δt denotes the duration of a time slot [s] and $u = \sqrt{\lambda} \cdot D$ is the dimensionless mean progress of a packet.

The dimensionless mean progress $u = u(N_R)$ is used as the performance measure for the forwarding capacity instead of I or D to eliminate the physical parameters dependent on the dimensions of the network from the results. The model used omits all the real-world phenomena that might be scale dependant, and instead, the results are presented as a function of the average size of a node's neighborhood N_R. Besides the simplicity, the convenience of $1/\sqrt{\lambda}$ as the unit length related to the model is based on the fact that the average distance between two nearest terminals is $1/(2\sqrt{\lambda})$ [21].

Because analytical results are mostly hard to achieve, the results are obtained from simulations. One possibility for estimating u is to monitor the total progress of

packets in a network with finite area. The mean progress of a packet D can be approximated from the total progress with the equation

$$D = \frac{1}{\tau N} \sum_i S_i, \tag{3.3}$$

where τ is the simulation time [time slots], N is the number of nodes, and S_i is the progress made by packet i. Alternatively, I can be interpreted as the average number of packets crossing a line of unit length perpendicular to the direction of progress. In this case, we can approximate the mean progress of a packet with

$$D = \frac{n}{\lambda \tau L} = \frac{n\sqrt{\pi}}{\tau M \sqrt{\lambda N_R}}, \tag{3.4}$$

where n is the number of packets crossing a line with the length L during τ time slots, and $M = L/R$ being thus the length of the line in transmission radii R. It is also to be noted that when the physical parameters of the network λ and Δt are fixed, the task of maximizing I equals the task of maximizing the dimensionless mean progress u instead.

3.3 Network as a graph

The exact modeling of the channel conditions of a wireless multi hop network is not a simple task. Many of the parameters are application dependent, and the wireless medium tends to vary dynamically. Modeling these things would make the model too complicated for our needs. Instead, we find it useful to abstract away the physical layer details of the network, and model the network as a graph. This kind of abstraction allows us to use existing mathematical tools and is beneficial when studying the general properties of the type of networks in question.

A graph G is a pair $G = (V, E)$ consisting of a nonempty set of vertices V and a set of pairs of distinct vertices, called the edges, E [48]. The vertices are also called nodes when we are talking about a graph that represents an actual network. In this case, the edges are referred to as links. The term network itself refers to a pair (G, c) of a graph and a mapping $c : E \to \mathbb{R}^+$. The number $c(e)$ is called the capacity of the link e. If there is no edge connecting two vertices the capacity may be set to zero.

For applications, especially those concerning traffic and transportation, it is often useful to give a direction to the edges of a graph. In a directed graph (digraph) the set E consists of ordered pairs $(u, v) \in V^2$ where $u \neq v$. As a distinction to the undirected case, the elements of E are called arcs. The names used with networks, nodes and links, remain the same, and the start vertex of a link $t(e) = u$ is called the transmitting node and the end vertex $r(e) = v$ the receiving node. Additionally, u and v are said to be incident with e.

Let us now consider a network (G, c) where G is directed, and we distinguish two special vertices: the start node s and the terminal node t, or the source and the sink, such that t is accessible from s. Accessibility means that there exists a sequence of vertices $(v_0, \ldots, v_n)$ (a walk) such that $(v_{i-1}, v_i) \in E$ for $i = 1, \ldots, n$, and $v_0 = s$ and $v_n = t$. Now we have a structure $\mathcal{N} = (G, c, s, t)$ that we call a flow network, and we can define a flow in the network. A mapping $f : E \to \mathbb{R}^+$ is a flow if it satisfies the following conditions:

1. $f(e) \in [0, c(e)] \quad \forall\, e \in E$
2. $\sum_{r(e)=v} f(e) = \sum_{t(e)=v} f(e) \quad \forall\, v \in V \setminus \{s, t\}$

The first, feasibility condition guarantees that there is a positive (≥ 0) bounded flow through every arc, and the second, flow conservation conditions means that flows are preserved (except at the source and the sink). The value of flow f is

$$w(f) = \sum_{t(e)=s} f(e) - \sum_{r(e)=s} f(e) = \sum_{r(e)=t} f(e) - \sum_{t(e)=t} f(e). \tag{3.5}$$

3.4 Interference

As mentioned in Section 3.1, a node is able to hear a transmission from any other node within its communication range. In our model, this means that there is a link between two nodes if they are within the distance R from each other, that is, $(u, v) \in E$ if $d(u, v) \leq R$. If there are two nodes transmitting within the communication range of a node, the transmissions interfere with each other and the node in question may not be able to receive one or either of them.

The effect of interference is described with the interference area of a link. The interference area $I(e)$ is the set of links that affect or are affected by the use of link $e \in E$. If link e is currently active, the attempt to activate any other link in $I(e)$ will result in a collision. When trying to use the network as efficiently as possible, the links that are active at the same time should not belong to each other's interference areas. If this happens, the reception of one or more packets will fail.

The interference model used to model collisions in this study is the Boolean interference model. According to the Boolean interference model, a node is only able to receive a packet if it hears exactly one transmission inside its transmission radius including its own, and thus

$$I_B(e) = \{a \in E \mid d(t(a), r(e)) \leq R \vee d(r(a), t(e)) \leq R\}. \tag{3.6}$$

The interference model and the assumption about the common fixed transmission radius together form what we call the Boolean model.[1]

[1] There exists a trivial extension for the model where the interference range is (usually) larger

Another possibility for the wireless model would be an interference model where both the transmitting and receiving node of the actual data packet may transmit (acknowledgement packets, control messages, etc.) and have to be able to receive during the time slot.

$$I_A(e) = \{a \in E \mid a \in I_B(e) \vee d(t(a), t(e)) \leq R \vee d(r(a), r(e)) \leq R\} \tag{3.7}$$

In practice, the amount of interference depends on various radio link properties like the number of transmitting nodes and their locations, background noise, etc. A node with a capture receiver may be able to receive the transmission with the strongest signal. There are several interference models that take these things into account [50]. For example, the capture threshold model considers the power ratio of the two strongest signals and the physical/additive models the signal-to-interference-and-noise-ratio (SINR). Even models based on actual measurements exist, but once again we try to keep the model general enough with minimal number of parameters.

If the network were, for example, a wired one, and the links did not interfere with each other, the interference area would be $I_0(e) = \{e\}$, if the nodes were able to transmit and receive multiple packets simultaneously, or

$$I_1(e) = \{a \in E \mid t(a) = t(e) \vee r(a) = t(e) \vee r(a) = r(e) \vee t(a) = r(e)\}, \tag{3.8}$$

if a node was only able to receive or transmit one packet in each time slot, and separate nodes did not interfere with each other.

3.5 Scheduling

According to the assumptions, all the nodes have equal properties, and since the channel quality and signal decay have been omitted, also the links have equal properties. This means that there exists a link between two nodes if the distance between them is less than the transmission radius R, and that all the links have the same capacity. This kind of network can simply be represented by a regular or directed graph $G(V, E)$.

The previous kind of model cannot be used to model flows, though, since the interference prevents one from using all the links at the same time. Instead, we have to establish a schedule $\boldsymbol{\alpha}$ which tells us how the links are used. All the links that are active simultaneously have to belong to the same independent set of links to avoid collisions. A set of links $\mathcal{L}$ is said to be independent under the Boolean model if

$$\forall a \neq e \ : \ a \notin I_B(e), \ a, e \in \mathcal{L}. \tag{3.9}$$

than the transmission range, but we omit this for simplicity. E.g., [49] separates transmission range, carrier sensing range, and interference range.

We call the independent sets that are used for transmitting transmission modes and denote the set of transmission modes with $\mathcal{M} = \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$. The schedule $\boldsymbol{\alpha} = \{t_1, \ldots, t_n\}$ assigns each transmission mode $\mathcal{L}_i$ with the proportion of time t_i that it is used. Now the capacity of link e is

$$c(e) = \sum_{i=1}^{n} t_i \mathbb{1}_{\{e \in \mathcal{L}_i\}}, \tag{3.10}$$

that is, the time share the link is active.

3.6 Connectivity

The performance of a wireless multi hop network is strongly dependent on its connectivity properties. The fraction of connected nodes describes the efficiency of the network and the level of connectivity of the nodes dictates the reliability.

Let $G = (V, E)$ be a graph. The set of neighbors of a vertex $v \in V$ is denoted by $N(v)$ and the number of neighbors $d(v) = |N(v)|$ is called the degree of the vertex. The average degree of G,

$$d(G) = \frac{1}{|V|} \sum_{v \in V} d(v), \tag{3.11}$$

is in other contexts referred to as the mean number of neighbors, and denoted with N_R when used as a parameter.

Two vertices s and t of graph G are called connected if there exists a walk $(v_0, \ldots, v_n)$, $\{v_{i-1}, v_i\} \in E \ \forall\, i = 1, \ldots, n$ with $v_0 = s$ and $v_n = t$. In a directed graphs the corresponding characteristic is called accessibility (see Section 3.3).

Graph G is called connected if any two vertices of G are connected. A directed graph G is called (strongly) connected if any vertex is accessible from any other vertex. We call a flow network connected if the corresponding digraph is connected, even if the capacity of a link belonging to a specific walk is zero under the schedule $\boldsymbol{\alpha}$ used. Thus, $\mathcal{N}$ is connected if any vertex is accessible from any other vertex under some schedule $\boldsymbol{\alpha}$.

A path is a walk $(v_0, \ldots, v_n)$ where the v_i are pairwise distinct. The number of disjoint paths has an important role on the network reliability. If the number of disjoint paths from s to t equals k, any $k - 1$ vertices (or edges) may be removed from the graph, and s and t still stay connected. A graph with k disjoint paths between every pair of nodes is called k-connected. The interdependence between k-connectedness and the minimum size of the separating set is summarized in the next theorem.

(Menger 1927) *Let $G = (V, E)$ be a graph and $A, B \subseteq V$. Then the minimal number of vertices separating A from B in G is equal to the maximum number of disjoint A-B paths in G.*

An A-B path is a path with $A \cap P = v_0$ and $B \cap P = v_n$, where $P = \{v_0, \ldots v_n\}$. A set of vertices X separates A and B if every A-B path contains a vertex from X.

3.7 Percolation

The original interpretation of the term percolation refers to the flow of fluids through random media. In networking, packets travel through a network similarly as a liquid flows through porous material, and thus, percolation describes the long range connectivity of the network. Percolation theory (see [51] for detailed definitions) provides a mathematical model for percolation, and it deals with the behavior of the connected clusters in a random graph.

The long range connectivity of the network is related to the existence of an infinite connected cluster – the so-called giant component. Let $G = (V, E)$ be a graph and $C(v)$ be the set of vertices that are accessible from $v \in V$. We denote the probability that $|C(v_0)| = \infty$ with $\theta(\boldsymbol{p})$, that is, the probability that an arbitrary node belongs to a cluster of infinite size, and call it the percolation probability. The percolation threshold is the critical surface for parameters $\boldsymbol{p}$ such that the percolation probability first becomes strictly positive, e.g., $p_c = \sup\{p \mid \theta(p) = 0\}$ for a single parameter. If $\theta(\boldsymbol{p}) > 0$, it now follows from Kolmogorov's zero-one law that there almost surely is some infinite cluster. The main results of percolation theory consider (prove) the existence of these critical values.

If the network (or the medium of the process) consists of the points of a regular lattice, the percolation threshold of the transmission radius is often trivial, e.g., the distance between two adjacent nodes in a square grid. The two types of percolation usually studied in case of a regular network are bond and site percolation. In the previous model, all the edges of the lattice belong, independently of each other, to E of $G = (V, E)$ with the probability p and do not with the probability of $1 - p$. In the latter model, all adjacent vertices of V are connected with edges, but the vertices of the lattice belong to V with the probability p. The critical probabilities for the most common regular lattices are listed in Table 3.1.

Table 3.1: Site (node) and bond (link) percolation thresholds for regular square, triangular, and hexagonal lattices.

Lattice	p_c^{bond}	p_c^{site}
square	$1/2$ [52]	≈ 0.5927 [53]
triangular	$2\sin(\pi/18) \approx 0.3473$ [54]	$1/2$ [55]
hexagonal	$1 - 2\sin(\pi/18) \approx 0.6527$ [54]	≈ 0.6970 [56]

As mentioned, percolation can be generalized to percolation on other graphs as well. The following theorem ([57], proofs in [51]) states that there exists a finite, positive value λ_c for the node density in our network model where the nodes are located

according to a 2-dimensional Poisson process and have a transmission radius of R, under which the percolation probability is zero and above which it is strictly positive:

Consider a Poisson Boolean model $\mathcal{B}(\lambda, R)$ in $\mathbb{R}^2$. There exists a critical density $\lambda_c > 0$ such that when $\lambda < \lambda_c$, all clusters are bounded a.s., and when $\lambda > \lambda_c$, there exists a unique unbounded cluster $\mathcal{U}$ a.s.

Instead of λ, the transmission range R can also be varied since the models $\mathcal{B}(\gamma\lambda, R)$ and $\mathcal{B}(\lambda, R/\sqrt{\gamma})$, where γ is a constant, are associated with identical graphs.

Despite the seeming simplicity of the model, the exact value of the critical density is not known. Some analytical bounds have been found (see, e.g., [51, 58]) in addition to several numerical estimates. For example [59] gives the estimate $\phi_c = 0.676339 \pm 0.000004$ for the critical volume fraction $\phi = 1 - e^{-\lambda\pi(R/2)^2}$ from which we get $N_R = -4\ln(1-\phi)$ for the mean number of neighbors and $N_R^c \approx 4.512$ for the percolation threshold.

The critical value for the mean number of neighbors is noteworthy when we later present the results for the forwarding capacity, since the true performance below the percolation threshold would always be zero. The infinite cluster guaranteeing the long range connectivity exists almost surely only in the super-critical phase ($\lambda > \lambda_c$).

Chapter 4

Maximum flow problem

4.1 Max-flow min-cut theorem

The maximum flow problem is a classic problem in graph theory and combinatorial optimization with a variety of applications. It considers finding a feasible flow through a flow network that is maximal. A flow f is maximal if $w(f) \geq w(f')$ for all flows f' on $\mathcal{N}$.

Before representing one of the fundamental results in flow theory, one definition is still needed. A cut of $\mathcal{N}$ is a partition $V = S + T$ where the plus sign denotes the union of two disjoint sets ($V = S \cup T$, $S \cap T = \emptyset$) such that $s \in S$ and $t \in T$. The capacity of the cut (S, T) is

$$c(S,T) = \sum_{t(e) \in S,\, r(e) \in T} c(e). \tag{4.1}$$

The cut $Q = (S, T) \in \mathcal{Q}$ is called minimal if $c(S, T) \leq c(S', T')$ for all cuts of the network. The minimum cut now has a significant effect on the capacity of the network.

(Ford and Fulkerson 1956) *The maximal value of a flow on a flow network $\mathcal{N}$ equals the minimal capacity of a cut in $\mathcal{N}$.*

This basically means that the bottlenecks of the network dictate the amount of traffic the network can carry.

4.2 Wireless model

With a fixed schedule the maximal flow in the wireless flow network equals the capacity of the minimal cut, but to find the overall maximum value for the flow, we also have to optimize the schedule. The value of the optimal flow ensues from the

problem

$$\max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}), \tag{4.2}$$

where $c(Q, \boldsymbol{\alpha})$ is the capacity of cut Q with schedule $\boldsymbol{\alpha}$, and $\mathcal{Q}$ is the set of all cuts.

Suppose we now have a large wireless network, and we study it under a fixed schedule $\boldsymbol{\alpha}$. Let $\mathcal{N}$ be the resulting flow network. The max-flow min-cut theorem states that the maximal value of a flow on $\mathcal{N}$ is equal to the minimal capacity of a cut in $\mathcal{N}$. The number of different cuts in the network is $2^{|V|-2}$, which makes finding the minimal cut an overwhelming task even for relatively small values of $|V|$. To find the overall maximum flow, we would still have to maximize the capacity of the minimal cut with respect to the schedule. Furthermore, to get some kind of insight about the maximal achievable flow with respect to the network density, we would also have to find the optimal transmission radius R. Changing the transmission range affects the underlying graph of the network, and the problem (4.2) has to be solved for each network (graph) separately. As a whole, this corresponds to solving the problem

$$\max_{R} \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}; R). \tag{4.3}$$

The above task of finding the exact forwarding capacity of a large wireless network is infeasible with the current methods, and thus, we seek to find upper bounds for the performance instead of trying to solve (4.3) exactly.

According to (4.2) the maximum with fixed R can be obtained by maximizing the minimum capacity of a cut with respect to the schedule. We can get an upper bound for the performance by limiting our examinations to a smaller set of cuts $\mathcal{Q}'$, because the minimum of a subset is always greater or equal to the original minimum. This gives us constraint

$$\begin{aligned} w(f_R^*) &= \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}) \\ &\leq \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}' \subset \mathcal{Q}} c(Q, \boldsymbol{\alpha}) \end{aligned} \tag{4.4}$$

for the maximum value of the flow. Another upper bound can be obtained by switching the order of minimization and maximization. Since the capacity of a cut with the optimal schedule, $c(Q, \boldsymbol{\alpha}^*)$, is always less or equal to the maximum capacity of the cut, $\max_{\boldsymbol{\alpha}} c(Q, \boldsymbol{\alpha})$, we have

$$\begin{aligned} w(f_R^*) &= \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}) \\ &\leq \min_{Q \in \mathcal{Q}} \max_{\boldsymbol{\alpha}} c(Q, \boldsymbol{\alpha}). \end{aligned} \tag{4.5}$$

The maximum capacity of a cut actually equals the size of the maximum independent set of links crossing the cut, because the maximum can be achieved by selecting as many independent links as possible and using them the whole time. This is always more than the capacity of the cut with the optimal schedule (for the whole network)

since the links cannot be used continuously in order for the flow network to be connected under that schedule (unless the link connects a source and a sink directly). Thus, the value of the maximum flow is limited by the size of the smallest maximum independent set of links crossing a cut in $\mathcal{Q}$.

Since the model used (see Section 3.1) only examines relay traffic, for example from the left side of the network to the right side of the network, we need to revise the definitions of the source and the sink. The analysis of relay traffic means that in our problem there would be a need for various sources and sinks instead of just one of each, because the traffic needs to be evenly spread. One formulation that saves us from this complication is the use of an augmented network with an artificial source and sink. The source is connected to all nodes within a range of R from the left side of the network with uncapacitated (wired) links that do not interfere with the other links, and the sink to all the nodes within R from the right side of the network. Because the links have infinite capacity, the minimal cut always goes through the original network.

4.3 Regular networks

When the network in question is very simple or the structure of the network is regular enough, it is sometimes possible to conclude the exact forwarding capacity of the network. We are now going to take look at the optimal schedules for some of the simplest regular lattices that include square, triangular, and hexagonal lattices. Occasionally, the hexagonal lattice is also referred to as the honeycomb lattice. The results are given for networks with same node density ($\lambda = 1$), and the correspondences between the node density and the distance between two adjacent nodes are given in Table 4.1. An additional assumption is that the direction of the packet flow is one of the main axes of the lattice.

Table 4.1: Node density versus the distance between adjacent nodes for regular lattices.

Lattice	node density	dist btwn nodes
square	$\lambda = \frac{1}{d^2}$	$d = \frac{1}{\sqrt{\lambda}}$
triangular	$\lambda = \frac{2}{\sqrt{3}d^2}$	$d = \sqrt{\frac{2}{\sqrt{3}\lambda}}$
hexagonal	$\lambda = \frac{4}{3\sqrt{3}d^2}$	$d = \frac{2}{\sqrt{3\sqrt{3}\lambda}}$

4.3.1 Square grid

The simplest regular lattice is the square grid. Since there is exactly one node in each square whose sides equal the distance between the nodes d, the dimensionless mean

progress of a packet is the portion of nodes transmitting multiplied by the progress of the transmission [d]. For example when the transmission range equals the distance between adjacent nodes, i.e., $R = d$, we have $u = \sqrt{1/d^2} \cdot 1/3 \cdot 1 \cdot 1d = 1/3 \cdot 1 = 1/3$ (see Figure 4.1 the first subfigure and Figure 4.2). When the transmission range grows to $\sqrt{2}d$, the links on top of each other start to interfere, and we would lose a half of the transport capacity if we continued forwarding in the direction of the packet flow. By transmitting obliquely, we get the dimensionless mean progress of 0.2. The progress of the packets remains the same, but we are now able to use one in five nodes to transmitting (Figure 4.1 the second subfigure).

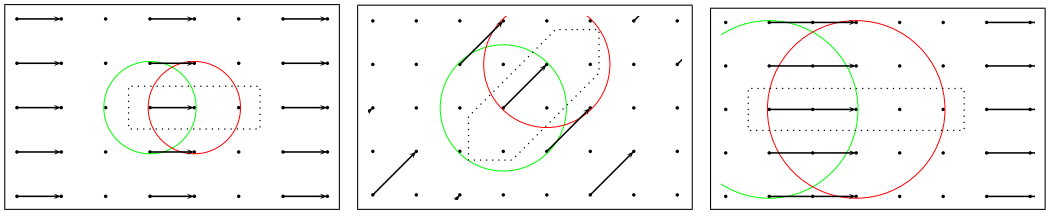

Figure 4.1: Examples of the independent sets of links of the optimal schedules for square grid with values $R = d$ (left), $\sqrt{2}d$ (center), and $2d$ (right figure). The repeating structures are illustrated with dotted lines.

The loss of forwarding capacity happens always until R reaches a multiple of d. The characteristic of the square grid is that when $R = nd$, $n \in \mathbb{N}$ the dimensionless mean progress gets notably higher values than otherwise, and it approaches $1/2$ when $n \to \infty$. This is due to the fact that the link upon another never interferes with it though the tolerance gets smaller and smaller as $\sqrt{(nd)^2 + 1} - nd \to 0$, when $n \in \mathbb{N}$ grows. The narrowing of the margin can be seen from the first and the last subfigure of Figure 4.1. In the last subfigure the illustrated transmission radius gets much closer to the receiving nodes of the neighboring links than in the first subfigure. Because an exact transmission range is not at all realistic in practice, other values around 0.2 give a better baseline for general comparisons.

4.3.2 Triangular grid

In a triangular lattice, each node has six neighbors, which makes the triangular lattice more vulnerable to interference with the shortest transmission radii. The difference to the square lattice can be seen in Figure 4.2. The most efficient way of sending packets is to place the links on top of each other as closely as possible in a similar fashion as with the square grid, but because of the different nature of the lattice, the towers emerging when $R = \sqrt{3}nd, n \in \mathbb{N}$ are tilted.

To be more exact, when the transmission radius equals $\sqrt{3}d$, it is possible to transmit between the two farthest corners of a diamond formed by two triangles. These links are part of a rectangular lattice with an extra node in the middle of each rectangle.

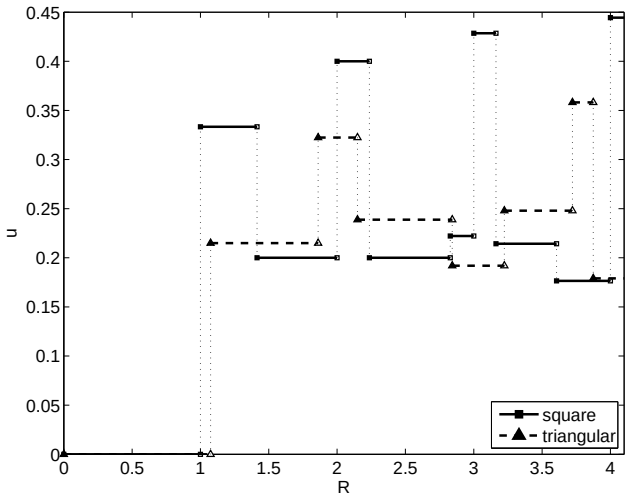

Figure 4.2: The dimensionless mean progress of a packet as a function of the transmission radius for square and triangular lattices. The transmission radii are given for a network with $\lambda = 1$ (see Table 4.1).

The same kind of structure appears also when transmitting in the direction of the packet flow, but instead of transmitting along the short edge like when $R = d$, we now transmit along the long edge.

The difference to the square case is that the extra nodes form a similar overlapping structure, which makes it possible to bring the towers a little closer to each other. This means that compared to the every third line of nodes in the square grid with $R = d$, every fifth line of nodes can transmit in triangular grid when $R = d$ or $R = \sqrt{3}d$. The dimensionless mean progress of a packet is thus $\sqrt{\lambda} \cdot \frac{1}{5} \cdot d \approx 0.22$ for the case $R = d$ and $\sqrt{\lambda} \cdot \frac{1}{5} \cdot \frac{3}{2}d \approx 0.32$ for $R = \sqrt{3}d$.

4.3.3 Hexagonal grid

The hexagonal lattice differs from the square and triangular lattices, because the nodes are not in symmetric positions. If a node has a neighbor in the direction of the packet flow, the next node in that direction does not have one. This makes finding the optimal schedule more difficult with hexagonal lattice, since all the links are no longer aligned, like Figure 4.3 compared to Figure 4.1 shows. Capacity-wise, this means that a hexagonal lattice network under the Boolean interference model is not that efficient since the links cannot be fitted close to each other. The concluded dimensionless mean progresses of a packet are represented in Figure 4.4.

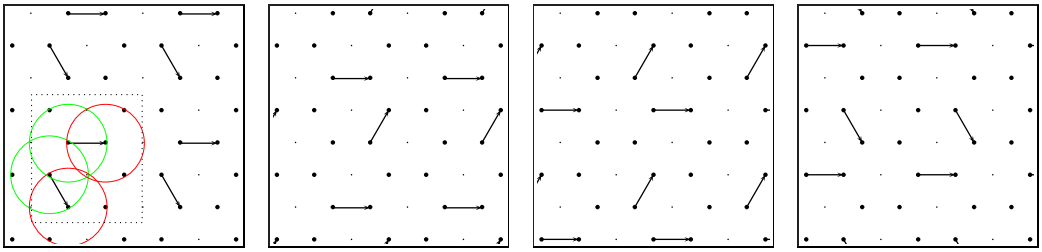

Figure 4.3: What seems to be the optimal schedule for hexagonal lattice with $R = d$. In contrast to earlier, all the links are not aligned, and the link formation differs from independent set to another.

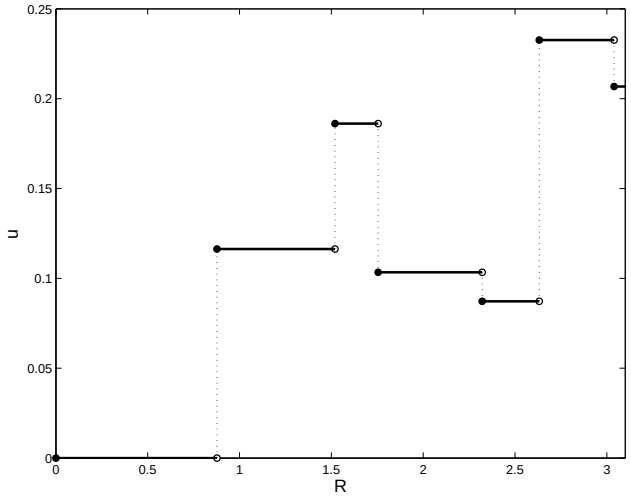

Figure 4.4: The dimensionless mean progress of a packet as a function of the transmission radius for hexagonal lattice. The transmission radius is given for a network with $\lambda = 1$ (see Table 4.1).

Chapter 5

Upper bound methods

5.1 Moving window algoritm

Because the mean density of progress, I (3.2), can be interpreted as the average number of packets crossing a unit length of line perpendicular to the direction of the flow in unit time, it is reasonable to consider cuts that correspond to a straight line in the vertical direction as the limited set of cuts $\mathcal{Q}'$ in (4.4). If the limited set of cuts additionally consists of a single cut, the task equals finding the size of the maximum independent set of links crossing the given line. Each independent set of links has a certain number of links crossing the cut, and maximizing with respect to $\boldsymbol{\alpha}$ picks up the one with the greatest value. In this case, the dimensionless mean progress can be approximated with the help of (3.4).

5.1.1 Algorithm

The problem is to find the maximum independent set of links crossing an arbitrary line in the infinite network. This is done with an algorithm similar to Retrospective optimization [60] by moving a window separating the nodes above and below along the line. A binary tree represents all the possible link combinations in the window area. The value assigned to each leaf is the size of the maximum independent set so far given the combination of active links in the window. Because the entering and exiting links are independent, we can combine the on- and off-branches corresponding to a link that has been dropped out of the window and choose the optimal (max) values for the new tree. This way we can recursively find the size of the maximum independent set of links. The Moving window algorithm (MWA) limits in no way the length of the simulation, and when the execution is continued, the result converges unbiasedly towards the true value.

Let us consider a window with the width of two transmission radii R, the height

of $3R$, and the line representing the cut going straight through the middle of the window in vertical direction (see Figure 5.1 for a picture). Set theoretically the window is represented with a set of nodes W while the set of links crossing the line in the window area is denoted with L. Considering simulation step i, we have the following definitions (l and r refer to the left and right sides of the network):

$$\begin{aligned} W_i &= W_i^l \cup W_i^r, \\ W_i^l &= \{v \in V \mid x(v) \in [x_0, x_0 + R],\ y(v) \in [y_i, y_i + 3R]\}, \\ W_i^r &= \{v \in V \mid x(v) \in (x_0 + R, x_0 + 2R],\ y(v) \in [y_i, y_i + 3R]\}, \\ L_i &= \{e \in E \mid t(e) \in W_i^l \wedge r(e) \in W_i^r\}. \end{aligned}$$

Widening the window would not increase the number of links in L, since no node to the left or to the right of the window is able to form a link with a node that is located on the opposite side of the line. At the same time, all the links with at least one end below the window (e_b) are always independent from the links with at least one end above the or at the upper bound of the window ($e_a \notin I_B(e_b)$). The total length of such two links is always less than $2R$ leaving more than one R between them as a sufficient margin.

All the possible combinations of links in L are represented by a rooted binary tree T. A tree is a connected graph that does not contain any cycles. A cycle is a sequence of nodes $(v_0, \ldots, v_n, v_0)$ where $(v_0, \ldots, v_n)$ is a path and $(v_n, v_0) \in E$. In a binary tree each vertex has the degree of three or less. We distinguish one of the vertices and call it the root r. The root can have only two neighbors. A vertex, other than the root, is a leaf if its degree is one.

Every level of the tree, i.e., vertices at the same distance from the root, (except the root itself) corresponds to a link in the window, and every edge describes the on-off state of that link in the link combination represented by the vertex incident with the edge and further away from the root. The value assigned to each vertex represents the size of the maximum independent set of links crossing the line so far given the combination of links determined by the vertex. Depending on the order in which the links are removed from the window, the value may or may not include links that were formed after the link corresponding to that level and that have been removed from the window. This means that only for the values assigned to each leaf (leaves describe the on-off state of each link in the window) are we able to give a simple mathematical expression.[1] Nevertheless, by looking at the tree depicted in Figure 5.1, we see how each link in the window is either on (A) or off ($\bar{A}$). The empty vertices in the figure have been added to illustrate the combinations that are impossible due to the interference. When one starts from the root and moves down

[1] Let i be the current step in the simulation, $G_i = (\bigcup_{j=0}^{i} W_j, \bigcup_{j=0}^{i} L_j)$ and $Q_i = (\bigcup_{j=0}^{i} W_j^l, \bigcup_{j=0}^{i} W_j^r)$. Each leaf, $\xi_i^k = (L_i^{on}, L_i^{off})$, of the binary tree T_i describes whether a link, $e \in L_i = L_i^{on} + L_i^{off}$, is on or off. If $e \in \mathcal{L}_j \cap L_i^{on} (/L_i^{off})$, then $t_j = 1(/0)$ in schedule $\boldsymbol{\alpha}$. The value assigned to a leaf ξ_i^k is thus $\max_{\boldsymbol{\alpha}|\xi_i^k} c(Q_i, \boldsymbol{\alpha})$.

the tree, the value assigned to a vertex increases every time one advances through a link that is on.

As mentioned earlier, the idea behind the algorithm is to move the window along the line. The algorithm consists of the steps presented in Table 5.1 or in more detail in Table A.1.

Table 5.1: Moving window algorithm.

0.	Initialize
1.	Draw step
2.	Drop nodes, remove links, update tree
3.	Add node, create links, update tree
4.	Goto 1.

During the initialization phase, all the variables including W, L, and T are initialized. The first actual step is to draw the location of the next node from the exponential distribution. Since the nodes are located according to a Poisson process, the vertical distance between two consecutive nodes in the path of the window is exponential with the parameter $2\lambda R$.

Step 2 is to remove all the nodes, whose vertical distance from the new node is greater than $3R$, from the window. Accordingly, all the links that the removed nodes are incident with are removed from L. Because these links cannot interfere with the links that become possible when the next node enters the window, they can also be removed from the tree T.

A link in the window corresponds to a level in the tree. Since the link that is being removed from the window cannot interfere with the links that enter the window later in the simulation, it is not necessary to know whether we are in the on- or off-branch corresponding to that link when adding a new link to the bottom of the tree. This means that we can compare the on- and off-branches and choose the maximum of the two in each vertex of the subtree to be the corresponding value (the value of the vertex above) in the updated tree. If the exiting link does not correspond to the top level of the tree, the same procedure is done to all the on-off pairs, and as a result, one level of vertices has been removed from the tree. The procedure is illustrated in the example of the following section 5.1.2.

Step 3 is to draw the horizontal location of the new node entering the window from uniform distribution and add the node to the set W. The new node now corresponds to the top of the window. If it is possible for the new node to form links crossing the cut, they are added to L, and also a new level of vertices is added to the bottom of the tree T for each new link. To every leaf of the tree, the off-branch is added without increasing the size of the maximum independent set. The on-branch with the increased value of the maximum independent set is added if the link combination is possible considering the interference.

If the sum of the steps exceeds the simulation length the simulation ends, otherwise the distance to the next new node is drawn.

5.1.2 Example

Figures 5.1-5.4 present an example of how the Moving window algorithm works. The example starts from a point where the window has already moved so that three links crossing the cut have become possible, but none has so far been dropped out from the window.

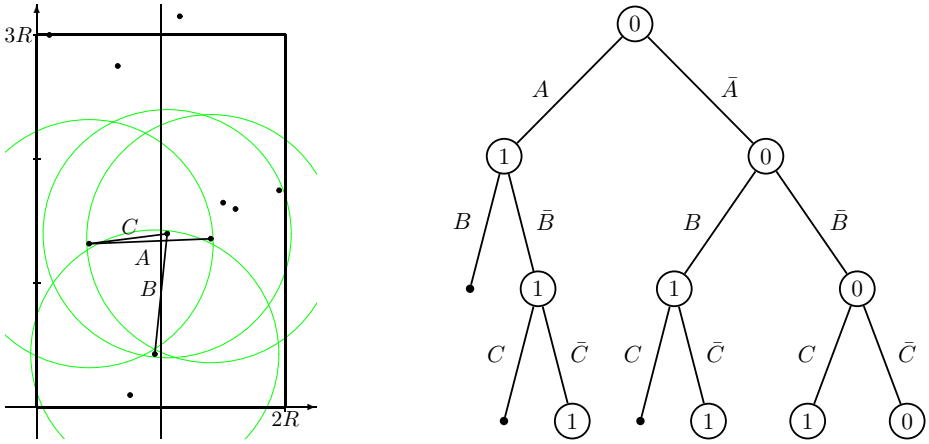

Figure 5.1: (Moving window algorithm example 1) At the beginning of the example there are three links in the window.

According to the description of the algorithm the next thing to do is to draw the distance from the top of the window to the next node. The next node has already been plotted to the Figure 5.1 for clarity, but so far we would only know its distance from the window. Next we remove all the nodes that are closer to the bottom of the window than this distance. This means that the one node in the bottom of the window is removed, but since it is not incident with any links no further action is needed.

Next step is to add the new node to the window. At this point we also fix the horizontal location of the node. The entering node makes a new link crossing the cut possible, and since it does not interfere with any other links, both on- and off-branches are added to each leaf of the tree. The value assigned to the new leaves in the on-branches is one greater than in the vertices above. The situation after step 3 is presented in Figure 5.2.

During the next round of the simulation, the transmitting node of link B is dropped out from the window, and the optimization procedure is initiated. The optimization procedure compares the corresponding vertices in the B- and $\bar{B}$-subtrees and chooses the greater to be the corresponding value in the updated tree. Figures 5.3 and 5.4

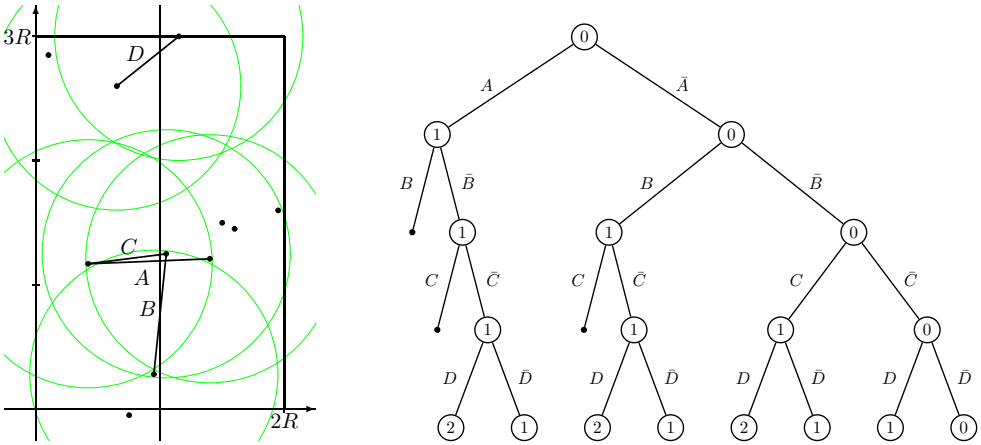

Figure 5.2: (Moving window algorithm example 2) A new node enters the window, and a new link is formed. (A node is also dropped out of the window, but no links are cut.)

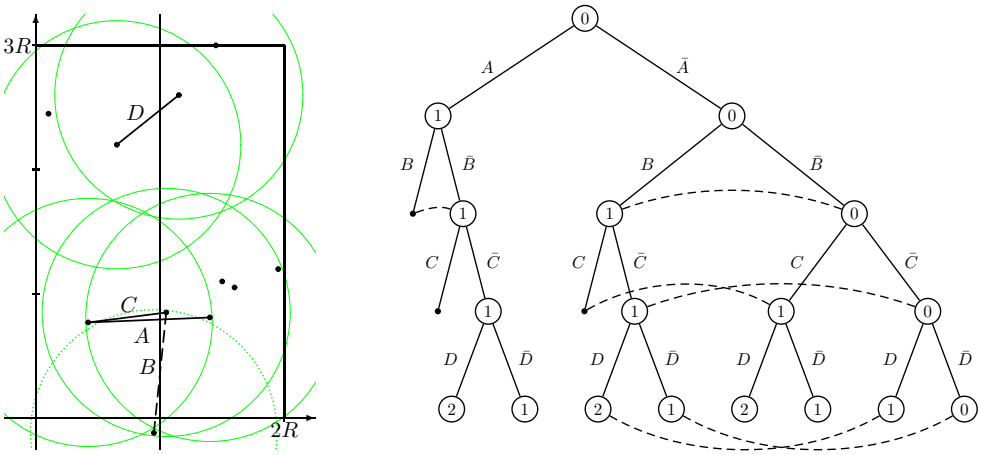

Figure 5.3: (Moving window algorithm example 3) The originating node of link B has been dropped from the window, and the optimization procedure has begun. The corresponding nodes in the B on- and off-branches are compared and the one with larger value is chosen for the new tree in Figure 5.4.

illustrate the situation before and after the level corresponding to link B has been removed from the tree.

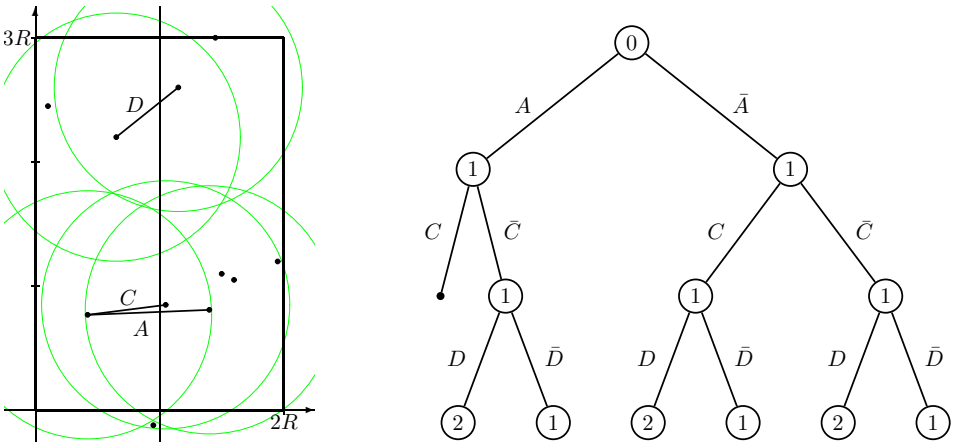

Figure 5.4: (Moving window algorithm example 4) The situation after the procedure.

5.1.3 Results

Since there are no nodes below the starting point of the window or above the ending point, the interference at the beginning and end of the simulation is lower than it should be. This means that the size of the independent set of links crossing the line is unnecessarily large. Compared to the situation with more interference the possible effect is only around one link too much at the bottom and one link at the top at most. If the new interference would cause the need of turning off more than one link, it would be possible to turn off just the interfering link, and the result would be just one link again. In theory, also larger values are possible, but in the simulations the effect was always less than one link, and as can be seen from Figure 5.5 the harmful border effect quickly becomes negligible when the duration of the simulation grows.

Figure 5.6 shows the dimensionless mean progress of a packet, $u(N_R)$, as a function of the average number of neighbors per node, N_R, for Moving window algorithm with the corresponding 90% confidence intervals. The third degree polynomial fitted to the data of the right subfigure gives the optimal value $u^* = 0.461$ with ${N_R}^* = 21.6$.

5.2 MWA with two cuts

The value for the dimensionless mean progress received from the Moving window algorithm is a relatively high upper bound for the forwarding capacity since the performance can only be achieved very locally. Although the cut is chosen to be arbitrary, after the maximization is done, it does not represent an average cut any-

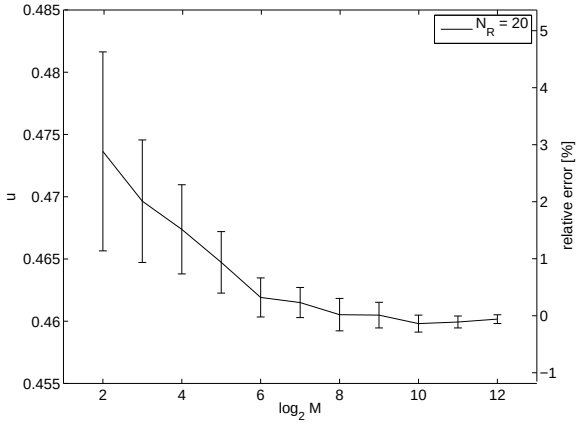

Figure 5.5: The dimensionless mean progress of a packet, $u(N_R)$, as a function of the base 2 logarithm of the simulation length M with corresponding 90% confidence intervals. The right vertical axis gives the relative error compared to simulations with $M = 10000 \approx 2^{13}$.

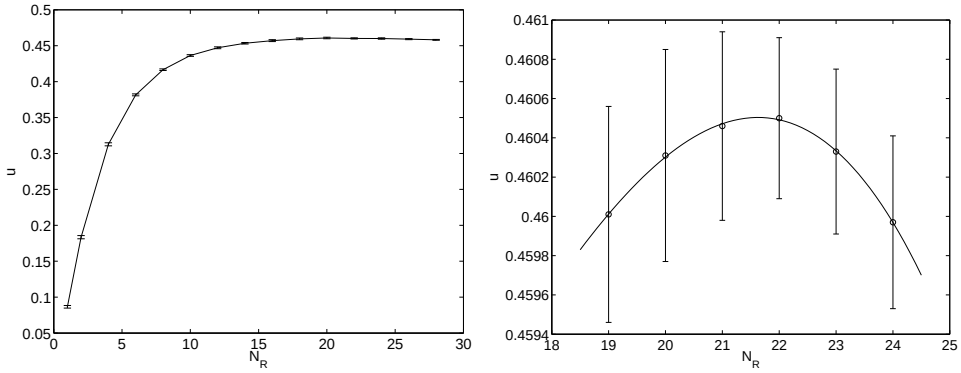

Figure 5.6: The dimensionless mean progress of a packet, u, as a function of the average number of neighbors, N_R, for Retrospective optimization with the corresponding 90% confidence intervals. The values of the subfigures are averages over 10 (left) and 25 (right) simulations with $M = 10000$.

more, because the same kind of performance cannot be achieved with a cut that suffers from the interference caused by the links of this cut. A natural way of improving the Moving window algorithm is to expand the limited set of cuts $\mathcal{Q}'$ in (4.4). A similar algorithm considering two cuts simultaneously (MWA_2) provides a tighter upper bound for $u(N_R)$.

When the number of cuts is two, the task is to maximize the smaller of the capacities of the two cuts. It is obvious that $\min\{c_1, c_2\} \leq (c_1 + c_2)/2$. Let us now consider the independent set of links that maximizes the number of times that the links cross the two lines. A schedule using only this independent set maximizes the sum in

the above inequality. Because the situation is symmetric and the lines are infinitely long, both lines are crossed equally many times. As a result the left hand side and the right hand side are equal, and as the right hand side was maximized, we have certainly obtained the maximum for the left hand side. The schedule is thus optimal.

Because we want the effect of horizontal interference to be included, the choice of the locations of the two cuts needs to be done carefully. If the cuts are far from each other, they are independent, and the result is the same as with the original method. On the other hand, if the cuts are very close to each other, almost all the links cross both of the lines, and because the links crossing both the lines are counted twice, the result is nearly the same again. On a large scale this means that the cuts need to be parallel. Thus, in order to maximize the effect of interference, we need to find the distance between the cuts, δ, that minimizes the performance in

$$\min_{\delta} \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}'_\delta} c(Q, \boldsymbol{\alpha}), \tag{5.1}$$

where $\mathcal{Q}'_\delta$ contains the two cuts on the distance δ from each other.

5.2.1 Results

Figure 5.7 shows the dimensionless mean progress of a packet, u, as a function of the distance between the two cuts, δ, for simulations made with the average of 12 neighbors per node.

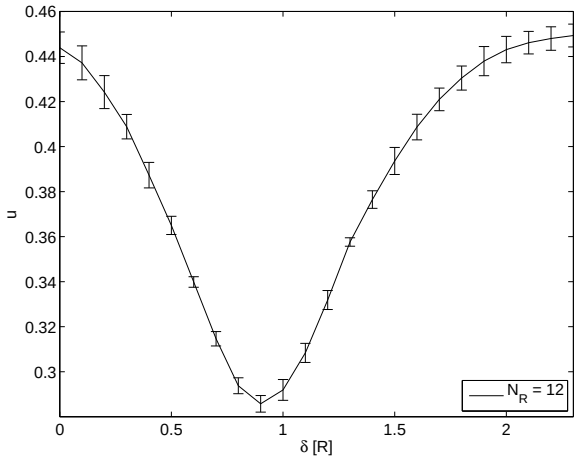

Figure 5.7: The dimensionless mean progress of a packet, u, as a function of the distance between the cuts, δ, for MWA with 2 cuts with the corresponding 90% confidence intervals. The values are averages over 5 simulations with $N_R = 12$ and $M = 1000$.

As can be seen from Figure 5.7, the value for u at the distance of zero is the same as with the original simulations with just one cut. When the distance between the cuts reaches $3R$, the value should again be the same as the links within the windows

cannot interfere with the links of the other window anymore. In practice this happens much earlier as the figure shows. The lowest dimensionless mean progress is achieved just before the distance of one, as expected. When the distance between the cuts is close to one, the number of nodes crossing both the cuts is very small, but the interference prevents one from substituting such links with two separate ones.

Based on the information of Figure 5.7 more simulations with distances just below one were made to determine the distance between the cuts that actually minimizes the maximum $u(N_R)$. The results from these simulations are presented in Figure 5.8. The search of the distance δ that minimizes the maximum of $u(N_R)$ equals the task of finding the saddle point of the hyperbolic paraboloid in the right subfigure of Figure 5.8. As can be seen from the figure, $u^*(N_R)$ is minimized when the distance between the two cuts is about 0.925.

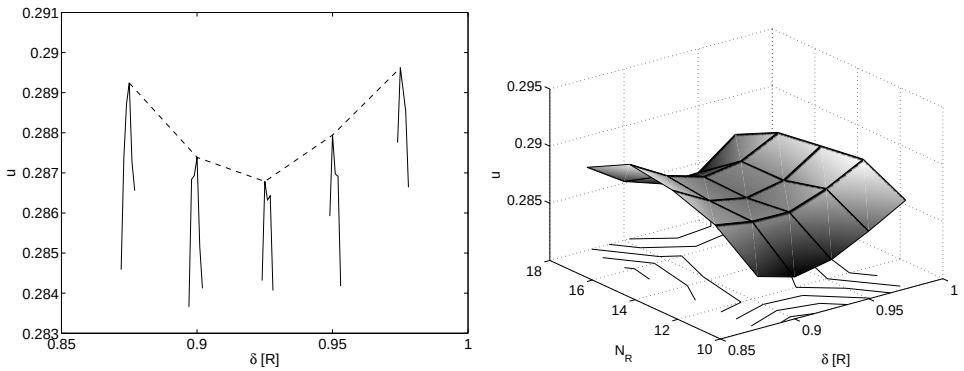

Figure 5.8: The maximum dimensionless mean progress of a packet, $u^*(N_R)$, as a function of the distance between the cuts, δ, for MWA with 2 cuts (dashed curve). The solid curves represent u for different values of N_R with the same δ. The right subfigure displays the same $u(N_R, \delta)$ in three dimensions. The values are averages over 10 simulations with $M = 1000$.

Finally, Figure 5.9 illustrates the dimensionless mean progress of a packet for MWA with two cuts with the distance 0.925 between them and the corresponding 90% confidence intervals. The results from Figure 5.6 have been added for comparison (dashed line). The optimal values with two cuts are $u^* = 0.287$ and ${N_R}^* = 12.5$ obtained from the third degree polynomial fitted to the data of the right subfigure.

5.3 MWA with infinite number of cuts

When the plane is filled with straight cuts, the task of maximizing the number of times a link crosses a cut becomes maximizing the sum of the progresses of the links, that is, finding the maximal weighted independent set. To clarify this, let n_i be the number of cuts the link e_i crosses. The number is proportional to the progress of the

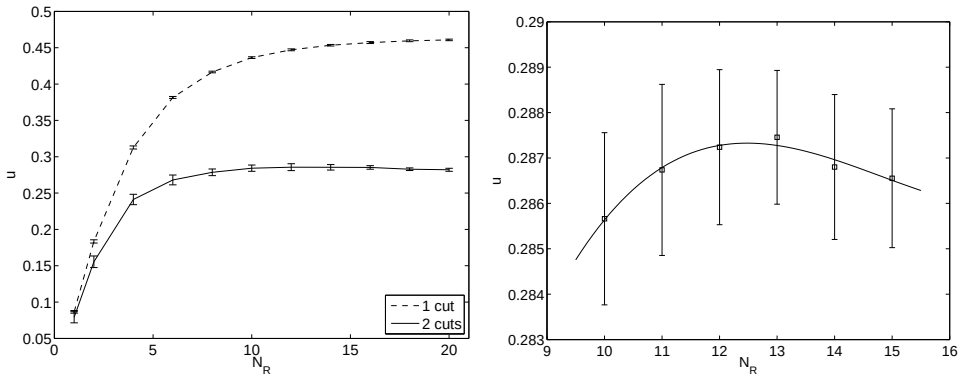

Figure 5.9: The dimensionless mean progress of a packet, u, as a function of the average number of neighbors, N_R, for MWA with 2 cuts (and for 1 cut with dashed line) with the corresponding 90% confidence intervals. The values are averages over 5 (left subfigure) and 15 (right subfigure) simulations with $M = 1000$.

link p_i, and we have

$$\lfloor \frac{p_i}{\delta} \rfloor \leq n_i \leq \lceil \frac{p_i}{\delta} \rceil,$$

where δ is the distance between the cuts. Thus, for the proportion n_1/n_2 we get

$$\frac{p_1/\delta - 1}{p_2/\delta + 1} < \frac{n_1}{n_2} < \frac{p_1/\delta + 1}{p_2/\delta - 1} \quad \Leftrightarrow \quad \frac{p_1 - \delta}{p_2 + \delta} < \frac{n_1}{n_2} < \frac{p_1 + \delta}{p_2 - \delta}.$$

So when the number of cuts per unit length goes to infinity, $\delta \to 0$ and $n_1/n_2 = p_1/p_2$. Thus, the contribution of a link is equal to its progress, and in order to get the optimal value for u, we need to find the independent set of links with maximal total progress.

The argumentation for the optimality is the same as in the two-cut case. The lines are in equal positions and infinitely long, and thus, by maximizing the sum, we maximize the minimum. Because the set of cuts is still a subset of all the cuts, the result is an upper bound for the sustainable flow. This is easy to see since the same progress cannot be obtained in each slot as the same independent set cannot be used repeatedly since the corresponding flow network is not even connected (the capacity of the minimum cut is zero). Actually, we get the maximal total progress that can be achieved in one time slot.

If we limit the width of the window for simulation purposes, we lose the symmetry since the cuts near the border are in unequal position. An infinite width is infeasible, but the harmful border effects in the horizontal direction can be diminished by connecting the opposite sides of the window together to form a tube. The perimeter of the tube still needs to be large enough to fit several consecutive links. This combined with the fact that none of the links can now be discarded straight away since they all cross a cut, place limitations on the simulation parameters and increase the memory requirements compared to the previous methods.

5.3.1 Practical simulation issues

In the original MWA, we chose the height of the window such that we were able to be sure that the links that were dropped out of the window did not interfere with the links entering the window. In practice, this means that there were often many links in the window that could have been removed earlier. For example, the links with both ends in the lowest third of the window always have at least the required R between them and the future links. To save memory, we want to remove such links in the MWA with infinite number of cuts.

Thus, instead of a piece of tube moving along a tube the new situation corresponds more to a case where a ring representing the upper bound of the window moves along a tube. The links kept in the binary tree are those with both ends below the ring (or at the same level as the narrow ring) that interfere with a link with one end above the ring level. The algorithm is very similar to the original algorithm with the difference that we need to keep track of links that enter the window in the near future, that is, have one end within a transmission radius from the window.

Although only the necessary links were kept in the binary tree, the size of the tree might still grow rapidly and become infeasible due to the stochastic nature of the processes generating the node locations. For this kind of situation we place a maximum for the size of the tree. When the number of nodes in the tree grows larger than this value, we start removing links from the tree even though they are still relevant. By removing the shortest links first we make sure that the error made is relatively small.

If we remove a link that still causes interference for the future links from the tree, it might become possible to add a link to such a branch of the tree where it would not really be possible. In that case the sum of progresses assigned to that leaf is unrealistically large. The size of the error is the progress of the removed link. If the

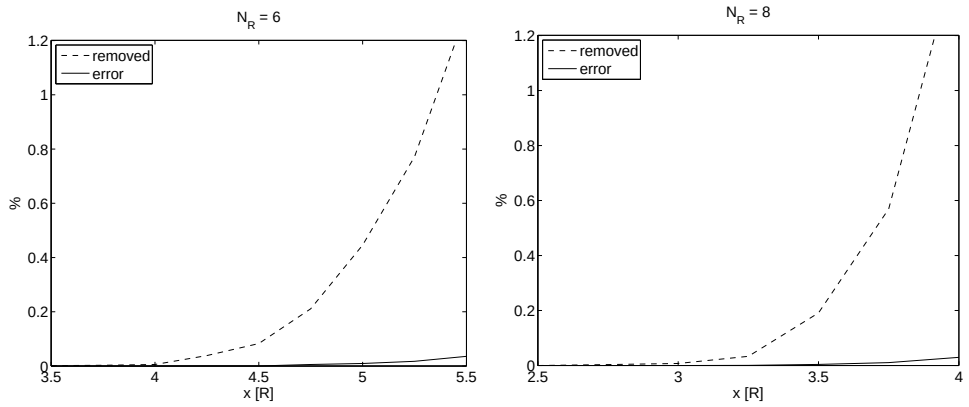

Figure 5.10: The total progress of the removed links and the real error for simulations made with $N_R = 6$ and $N_R = 8$.

link has a very small progress, the probability that the link is active in the optimal branch is small (if the link is not active in the optimal branch, removing will not cause any error), and even if the error is made the size of such an error is small.

The total progress of the removed links gives an upper bound for the error, but as Figure 5.10 shows, when the total progress of the removed links is around one percent of the total progress (the progress of the optimal set of links), the real error is less than 0.1%. This is also of the size of the biggest errors made in the simulations. For the comparison with the correct values to be possible, the simulations for Figure 5.10 were made with a tighter limitation on the size of the binary tree.

Figure 5.16 shows the data points obtained from the simulations. With neighborhoods from 7 to 12, the simulations with the one or two widest tubes would, at some point, exceed the 2GiB memory limitation of 32-bit architecture without the restricted binary tree. Other than that, the time is also a constraining factor, and additional datapoints would in many cases require over a month of computer time.

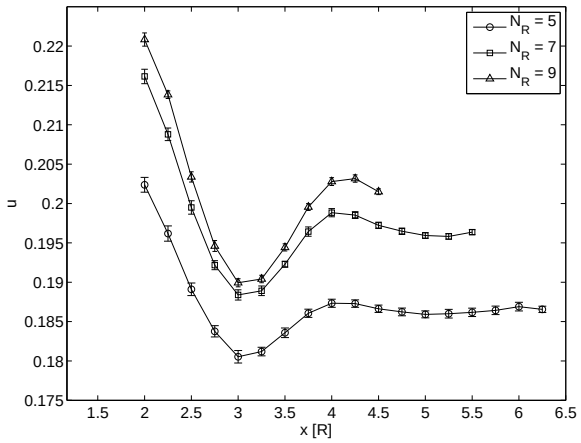

Figure 5.11: The dimensionless mean progress of a packet, u, as a function of the perimeter of the tube, x, for MWA with infinite number of cuts for three different values of N_R with the corresponding 90% confidence intervals. The values are averages over 20 simulations with $M = 1000$.

5.3.2 Results

Figure 5.11 shows the dimensionless mean progress of a packet, u, as a function of the perimeter of the tube, x, with the average of 5, 7, and 9 neighbors per node. As can be seen from the figure, u starts to stabilize only when the tube is wide enough to fit several consecutive links. The maxima appear when the tube is wide enough to hold full length links and the margins between them, that is, when the perimeter of the tube $[R]$ is even. The reason for this is in the formations the links tend to be placed that can later be seen from Figure 5.18. Because the locations of the nodes

are random, the maxima do not appear exactly with even values, but some tolerance is required.

Since the behavior of u seems quite regular, we try fitting a damped oscillating curve

$$u(x) = Ae^{-Bx}\cos(Cx + D) + E \tag{5.2}$$

to the data. Figure 5.12 presents the results of these fits for the average neighborhoods of 5, 7, 9, and 11. As can be seen from the figure the fits seem quite accurate. Figure 5.13 shows, though, that this is not fully true. When $N_R = 11$, we are only able to simulate data points around the first minimum. This is not enough to fix the positions of the minima and maxima, i.e., the frequency and phase terms of the curve, and they clearly differ from other curves.

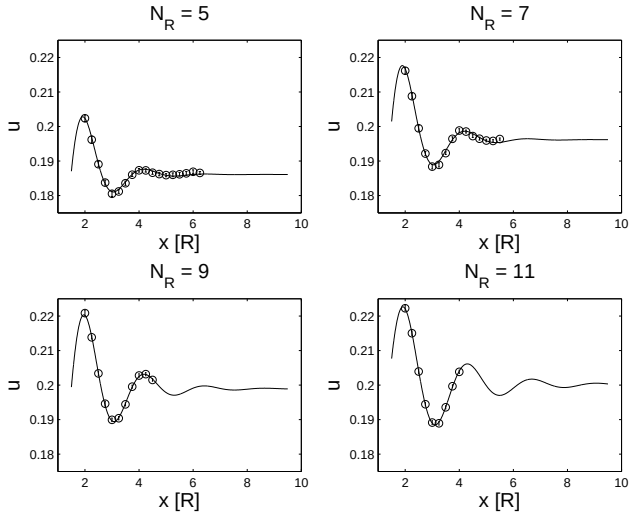

Figure 5.12: The dimensionless mean progress of a packet, u, as a function of the perimeter of the tube, x, for different values of N_R when (5.2) has been fitted to the data.

Figure 5.14 represents the parameters of (5.2) as a function of N_R. Based on the figure we replace the individual constant amplitude and damping terms with linear terms. From the single and two cut versions we also know that the constant term can be well approximated with a polynomial of third degree near the optimum. As mentioned, the frequency and phase terms should be constant (and we keep them such), but because of the randomness and the distance between the data points, as well as the scarcity of data with $N_R = 11$, some deviation is visible. Thus, we have the surface

$$\begin{aligned} u(N_R, x) &= (A_1 N_R + A_2)e^{-(B_1 N_R + B_2)x}\cos(Cx + D) \\ &\quad +(E_1 {N_R}^3 + E_2 {N_R}^2 + E_3 N_R + E4) \end{aligned} \tag{5.3}$$

to be fitted to the data. The resulting parameters from this fit are also plotted in Figure 5.14 (dashed lines). All the available data points were used, and the resulting

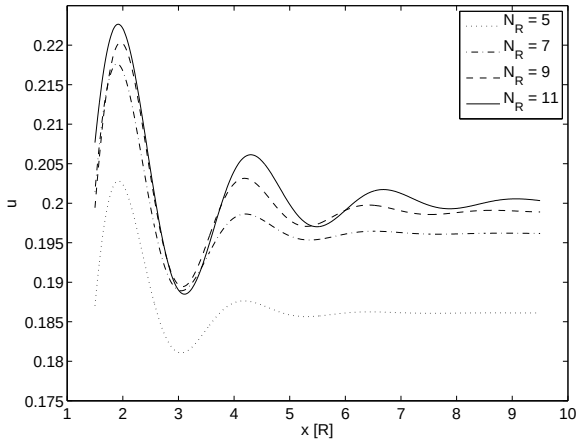

Figure 5.13: The dimensionless mean progress of a packet, u, as a function of the perimeter of the tube, x, for different values of N_R. The figure shows the anomalous frequency and phase parameters when $N_R = 11$.

surface is depicted in Figure 5.15. From this figure we obtain the optimal values $u^* = 0.20$ with $N_R^* = 9.9$. (Figure 5.16 shows the data to which (5.3) has been fitted in Figure 5.15 for obtaining the parameter presented in Figure 5.14 with dashed line.)

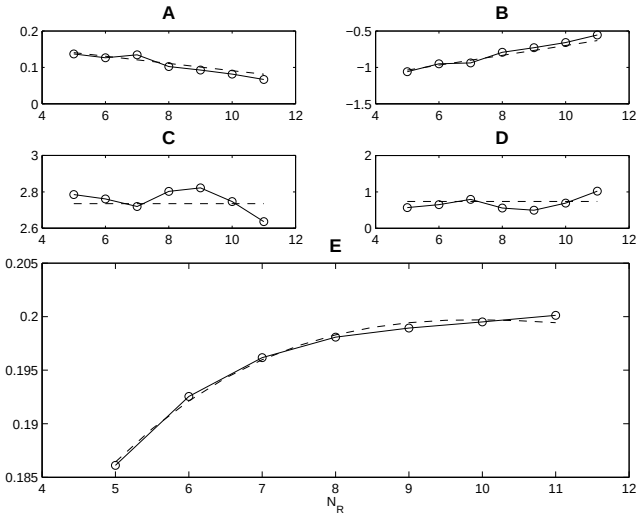

Figure 5.14: The parameters of (5.2) for different values of N_R when fitted to the data. The dashed lines give the corresponding values for (5.3).

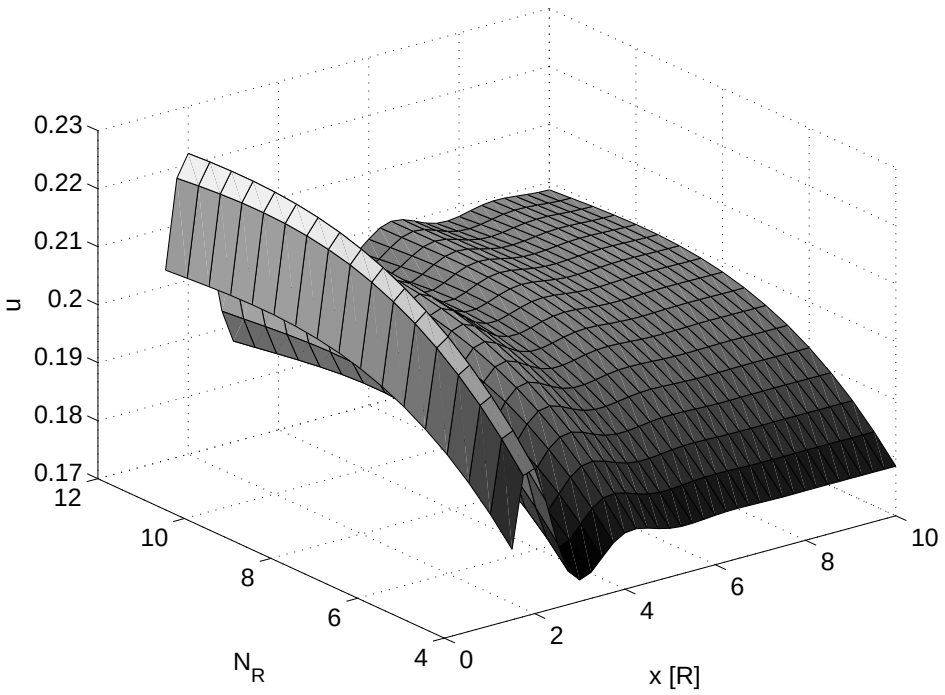

Figure 5.15: The dimensionless mean progress of a packet, u, as a function of the size of the average neighborhood, N_R, and the perimeter of the tube, x, when (5.3) is fitted to the data.

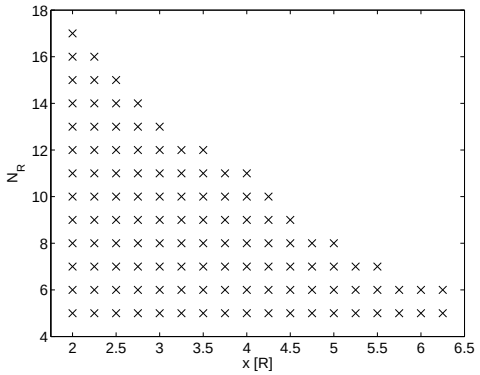

Figure 5.16: The data used for fitting (5.3) in Figure 5.15.

5.4 Approximative methods

5.4.1 Greedy method

An easy way to get an approximative upper bound for the forwarding capacity is to use a some kind of greedy method to find an approximation for the independent set with maximal total progress of the links. The maximal independent set gives an upper bound for the performance since the same progress cannot be achieved in consecutive time slots (see Section 5.3), and with the greedy method we actually get a lower bound for the capacity of the maximal independent set. This lower bound of the upper bound gives the approximation we require.

The simplest and literally greedy method is to choose links in order of their x-progress to the independent set for as long as links independent from the set are found. The greedy algorithm works as follows. Take a unit square to represent the supposed infinite plane and connect the opposite sides together to form a torus and to lower the effects of the borders. Place the nodes to the square and thus to the torus according to the two dimensional Poisson point process with node density λ. Gather all the links that are formed with the transmission radius R to a list and sort the list according to the progress of the link in descending order.

Go through the list and remove all the links that belong to the interference area of the first link excluding the first link itself. Then do the same for the second link in the list from which the links interfering with the first link have now been removed. Repeat this until all the links still remaining in the list have been gone through. Eventually, the list consists of links that do not belong to each other's interference areas, and we can calculate the approximation for the upper bound of the dimensionless mean progress with (3.4).

The results from the greedy method are represented in Figure 5.17. The simulations were made using the node density $\lambda = 2500$, and the values are averages over 100 simulations. Figure 5.19 displays the dimensionless mean progress for the average neighborhoods from 5 to 10. The third degree polynomial fitted to this data gives the values $u^* = 0.167$ with $N_R^* = 6.98$ for the optimum.

Figure 5.18 shows an example of how the links of the independent set from the greedy method are placed in the unit square. The left subfigure represents a simulation with $\lambda = 250$ and $N_R = 10$, and the right subfigure with $\lambda = 500$ and $N_R = 20$. The figures show that the links tend to be placed in a similar fashion as in the optimal schedules of the square lattice with $R = nd$ (see Figure 4.1). This is due to the chosen interference model, but it shows the benefit of transferring the packets in these wavelike formations. The towers that the active links form are visualized with the thick lines.

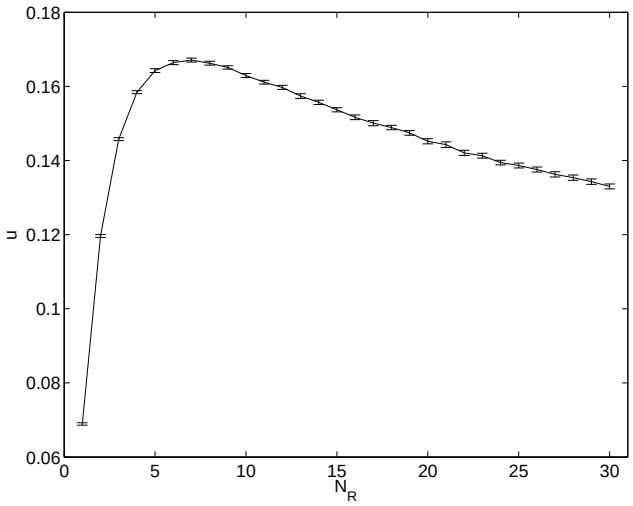

Figure 5.17: The dimensionless mean progress of a packet as a function of the average number of neighbors for the greedy method. The values are averages over 100 simulations with the corresponding 90% confidence intervals as error bars.

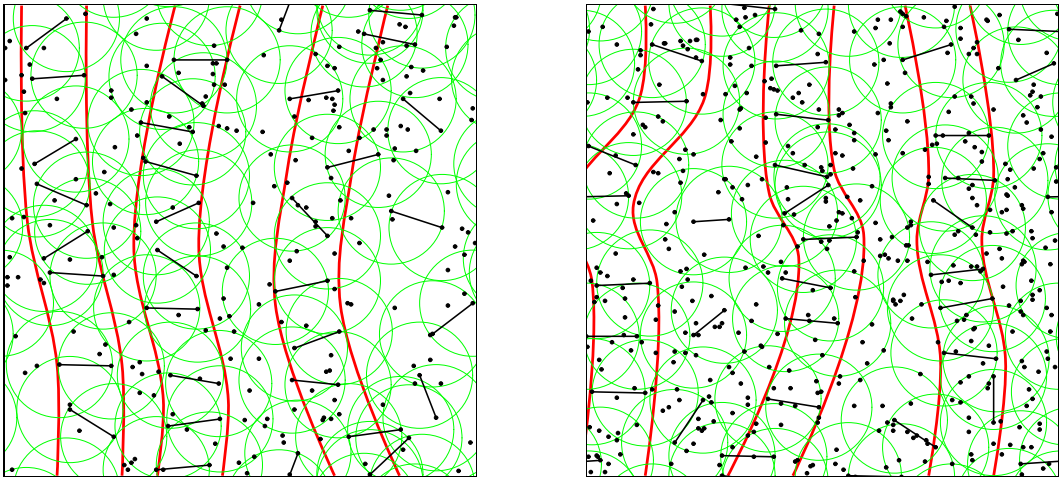

Figure 5.18: Two examples of the independent set of links resulting from the greedy method with $N_R = 10$ (left) and $N_R = 20$ (right figure). The links tend to form similar structures as the optimal schedule in the square grid. The thicker lines have been added merely to visualize this behavior.

5.4.2 Reverse greedy method

It is possible to present several heuristics for selecting the independent set of links relatively close to the optimal. We now take an opposite approach and try to remove the links that are somehow 'bad' instead of selecting the links that have the greatest progress.

Let L be the set of all links in the torus, and L^{on} the set of links that are on. We measure the 'badness' of link e with a value

$$b(e) = \frac{1}{p(e)} \sum_{a \in I_B(e) \cap L^{on}} p(a) - 1, \tag{5.4}$$

where $p(e)$ is the progress of link e. If the link interferes with links that have high total progress or is short, it is more likely that the link is removed. If the link does not interfere with any other links, $b = 0$.

The initialization phase of the algorithm is the same as in the greedy method, and in the beginning, all the links belong to the set L^{on}. After the initialization, the value b is calculated for all the links in L. The link with the greatest b is removed from L^{on}, and the values of b are updated to match the current situation. The link with the greatest b is removed until $b = 0$ for all the links in L^{on}. After this has been done, only links that are independent from each other are left.

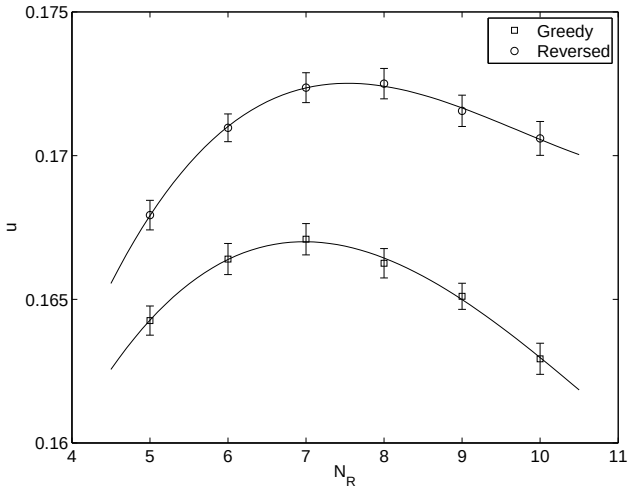

Figure 5.19: The dimensionless mean progress as a function of the average number of neighbors for greedy and reversed greedy methods near the maximum value.

Figure 5.19 shows the dimensionless mean progress achieved with the reversed greedy method as a function of the N_R with the corresponding 90% confidence intervals. The values are averages over 100 simulations with 2500 nodes. The fitted third degree polynomial gives the values $u^* = 0.173$ with $N_R^* = 7.54$ for the optimum. The obtained u^* is somewhat higher than with the basic greedy method.

It would be easy to find even better heuristic algorithms. For example, the independent set of links that the reverse greedy method produces is not necessary maximal since the method may remove links from the set L^{on} that later, when more links have been removed, become independent from the set. Anyway, these results provide us an adequate baseline for comparisons and we content ourselves with them.

Chapter 6

Forwarding methods

6.1 Existing results

So far, we have discussed about the theoretical and approximative upper bounds for the forwarding capacity of a wireless multihop network. To get some kind of idea how far these upper bounds are from actual performances, we are now going to take a look at some simulated forwarding methods [61, 62]. These results provide us a certain lower bound for the maximal performance.

In addition to the assumptions made in Section 3.1, the use of a MAC protocol similar to slotted ALOHA [4] is assumed. The protocol is characterized by a single parameter p which defines the probability that a node with queued packets transmits in a given time slot. Successful transmissions are acknowledged, but the time required for these messages is negligible. The parameter p is constant, which means that no backoff scheme is applied in retransmissions.

Finally, it is assumed that a node has the necessary information to be able to make the forwarding decision locally according to the rules in question. This means that a node knows at least its own coordinates, the coordinates of its neighbors, and the direction of the packet flow.

The simulation model [61] uses a unit square with an average of thousand nodes ($\lambda = 1000$) to represent the infinite network. The opposite sides of the square are connected together to form a torus, thus avoiding some of the harmful border effects. All the nodes with no neighbors in the direction of the packet flow are removed recursively[1], as recovering from them would require a specific procedure like face routing [23]. The traffic consists of packets with infinite life time circling around the torus for the simulation length, and the estimate for $D(N_R, p)$ is calculated from the total distance travelled by the packets with (3.3).

[1] $\mathbb{P}(\text{node concave}) = e^{-\frac{1}{2}N_R}$, which means that the number of these nodes in dense networks we are studying is very small.

The simulated methods include Most forward within radius (MFR) [21], Random forwarding (RF), Weighted random forwarding (WRF), and Opportunistic forwarding (OF) [61] as well as Modified weighted random forwarding (MWRF), Weighted expectation forwarding (WEF), and Current maximum expectation forwarding (CMEF) [62].

The MFR method simply chooses the neighbor with the greatest progress to be the next hop. This leads to poor utilization of the resources in static networks where the traffic paths are predetermined. The randomized methods (RF, WRF, MWRF, WEF) choose a certain neighbor with a probability proportional to, for example, the progress (see Table 6.1). This helps to spread the traffic more evenly to the network.

Table 6.1: The probability q_{ij} of becoming a receiver for a forward neighbor j when the sender node is i with randomized forwarding methods. The number of forward neighbors, i.e., neighbors with positive progress, is denoted with N_{F_i} and the number neighbors of node j with N_{R_j}.

	RF	WRF	MWRF	WEF
$q_{ij} =$	$\frac{1}{N_{F_i}}$	$\frac{d_{ij}}{\sum_{k=1}^{N_{F_i}} d_{ik}}$	$\frac{d_{ij}/N_{R_j}}{\sum_{k=1}^{N_{F_i}} d_{ik}/N_{R_k}}$	$\frac{d_{ij}(1-p)^{N_{R_j}}}{\sum_{k=1}^{N_{F_i}} d_{ik}(1-p)^{N_{R_k}}}$

The CMEF utilizes information about the activity of its surroundings. When the actual number of nodes that can possibly transmit inside each neighbor's transmission radius during the next time slot is known, it is possible to calculate and maximize the expected progress of the packet. The next hop j for node i is chosen to be

$$j = \arg\max d_{ij}(1-p)^{N_{A_j}-1+\mathbb{1}_{A_j}}, \tag{6.1}$$

where N_{A_j} is the number of active neighbors, i.e., neighbors with queued packets, of node j and $\mathbb{1}_{A_j} = 1$ if node j has queued packets and is zero otherwise.

The OF method demonstrates the benefits of local coordination. In OF a packet is broadcasted to all forward neighbors, and from the forward neighbors able to receive the packet, the one with the greatest progress is chosen to be the next hop. Thus, the receiving node

$$j = \arg\max\ d_{ij}\mathbb{1}_{R_j}, \tag{6.2}$$

where $\mathbb{1}_{R_j} = 1$ if neighbor j receives the transmission, i.e., there is no collision, and zero otherwise. Opportunistic forwarding is a variation of Extremely Opportunistic Routing (ExOR) [63], but it has been modified mainly to avoid duplicate packets.

The maximal dimensionless mean progresses and the corresponding parameters for the previous forwarding methods are given in Table 6.2 [62]. The greatest u was achieved with Opportunistic forwarding.

Table 6.2: The maximum $u(N_R, p)$ for each forwarding method along with the corresponding N_R and p.
*** The number of neighbors for each receiver is known.**
† The number of active neighbors, i.e., ones with queued packets for each receiver is known.
‡ The receiver with the best achieved progress is chosen after sending the packet to all receivers.

	$u(N_R, p)$	N_R	p
MFR	0.0126	50	0.35
RF	0.0222	14	0.25
WEF*	0.0253	16.5	0.21
WRF	0.0279	14	0.30
MWRF*	0.0297	13.5	0.34
CMEF†	0.0467	13.5	0.43
OF‡	0.0590	18	0.40

6.2 Additional simulations

The previous simulation model that uses a toroidal geometry to represent the infinite network has some notable drawbacks. The simulation time required to get rid of the initial transient is long. It takes time before the packets are distributed according to the desired stationary distribution instead of the uniform distribution in the beginning. It is also difficult to determine the necessary number of packets, and while a larger number allows theoretically better results, the transient gets longer as well. The duration of the initial transient is also a function of the transmission probability p, and simulations with a higher transmission probability may require longer simulation times than it would be apparent if the transient duration is determined using lower value of p.

Another drawback is that when the packets circle around the torus more than one time, they have to face the same bottlenecks more than once. This means that the shape of the torus might concentrate the traffic to a narrow band, leaving part of the network almost or completely unused.

Since we are interested in the maximal achievable capacity of the network, it is not appropriate to have these kinds of limitations that refer to a particular traffic situation instead of more optimal conditions.

To overcome the limitations the torus is opened into a tube. The top and the bottom of the unit square representing the infinite network are still connected to limit the border effect in the direction of the packet flow, but now the packets are being generated in one end of the tube and they travel through the network just once. To be more exact, the nodes within one transmission radius, R, from the left side are sources and they receive (generate) one new packet in every time slot they are

not transmitting nor hearing a transmission (i.e., are able to receive). The nodes within R from the right end of the tube are sinks. Instead of monitoring the total progress of the packets, it is enough to observe the number of packets that the sinks receive per time slot, that is, the packet flow. Based on that, the dimensionless mean progress can be calculated using (3.4). Because all the packets entering the sinks have roughly the same progress, i.e., the width of the square/the length of the tube, the result is equivalent (apart from the initial transient) with the one that would be obtained from (3.3).

Since the network is empty at the beginning of the simulation, is takes some time before the packet flow to the sinks has stabilized. This initial transient is depicted in Figure 6.1 (the depicted values have been simulated with near-optimal parameters). As can be seen from the figure, the packet flow reaches the stationary level quite fast and the deviation thereafter is relatively small.

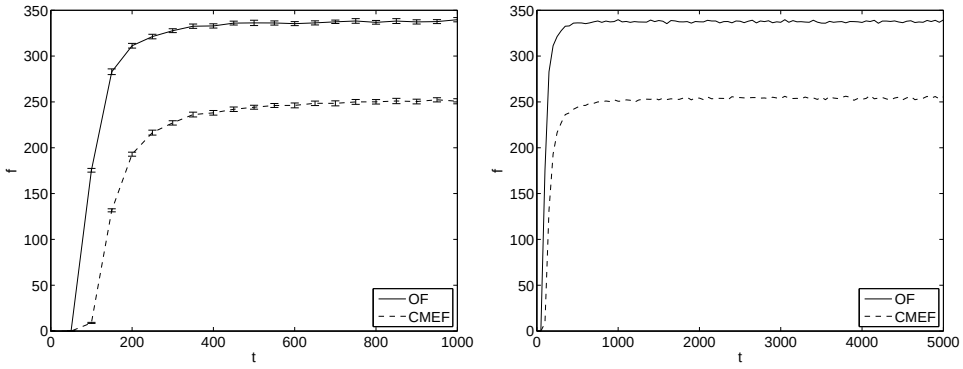

Figure 6.1: The number of packets received by the sinks, or the packet flow through the network, for CMEF and OF in a network with $\lambda = 10000$. The values are averages over 50 simulations, and the error bars of the left subfigure represent the 90% confidence interval.

The model also suffers from border effects since the nodes in each end experience the interference differently than the nodes in the middle of the tube. Especially the nodes near the sinks have less interference since the sinks do not transmit at all. These border effects can be diminished by making the tube longer. It is to be noted, though, that every finite network (with nodes distributed according to the two dimensional Poisson point process) has a positive probability to be disconnected. Increasing the length of the tube only, will thus decrease the capacity, not only because of the increased disconnecting probability, but also because the nodes will encounter more bottlenecks in general. This means that the height of the square forming the tube should be increased as well as the width as we approach the infinite network.

Figure 6.2 shows the dimensionless mean progress of a packet, u, as a function of the lenght of the tube, x. When the area from which the tube is formed is square, u approaches the true value. This is also the case for an area taller than the square

(perimeter of the tube $4x$). When the perimeter of the tube is constant (1), u is decreasing. When $x = 40$, the dimensionless mean progress is only about 85% from the real value with OF. With CMEF this happens already at $x = 20$.

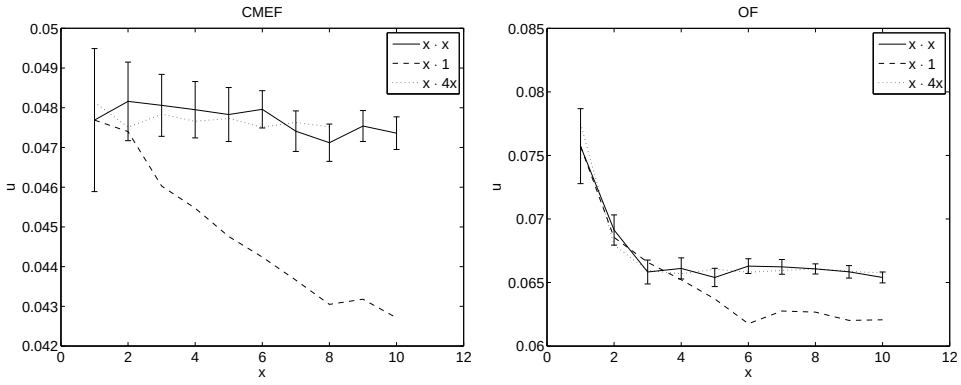

Figure 6.2: The dimensionless mean progress of a packet, u, for CMEF and OF as a function of the length of the tube, x, for cases where the perimeter of the tube is either same as the length, exactly one, or four times the length of the tube. The values are averages over 100 simulations with $\lambda = 100$ and near-optimal N_R and p. The error bars represent the 90% confidence intervals for the case where area is a square.

The simulations were made with a square network of 10000 nodes and the total simulation time of 20000 time slots, half of which were not used in calculating the results (transient duration). Figure 6.3 shows the dimensions mean progress, u, as a function the transmission probability for different values of N_R. Based on the third degree polynomial fitted to the data, the optimal transmission probability of CMEF with $N_R = 12$ is $p^* = 0.65$ resulting in the dimensionless mean progress of $u^* = 0.047$. The corresponding values for OF are $p^* = 0.77$ and $u^* = 0.065$ when the mean number of neighbors, $N_R = 19$.

As can be seen from the Figures 6.3 and the Table 6.2, the maximal dimensionless mean progress of a packet has increased, but the differences in the transmission probabilities seem more remarkable. The results are congruent with the assumption made earlier. If the initial transient in the original simulations is too short, they favor lower values of p. Also, if the traffic is more evenly spread though the network, the performance should be better.

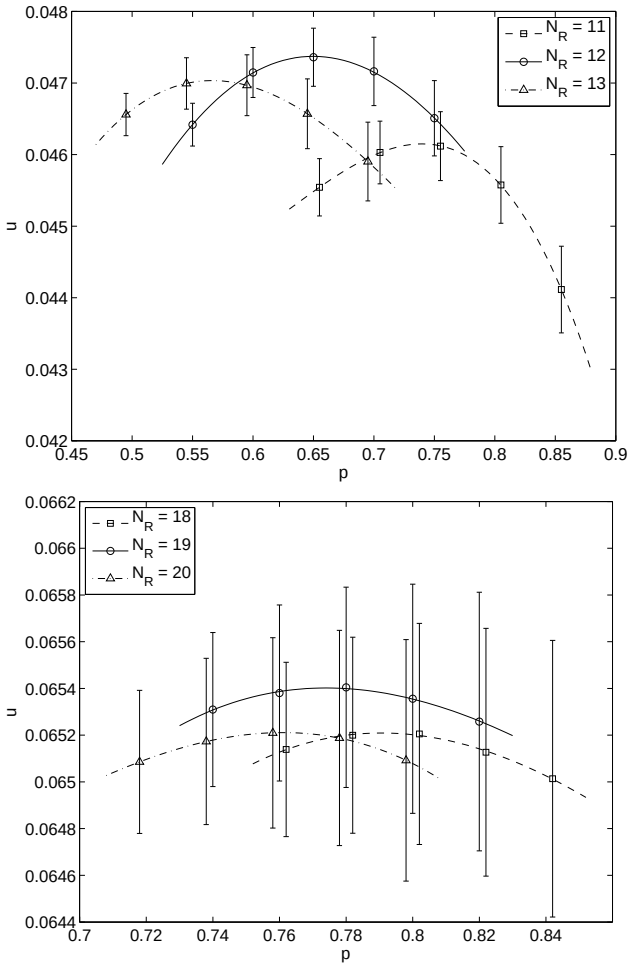

Figure 6.3: The dimensionless mean progress of a packet, $u(N_R, p)$, as a function of the transmission probability, p, for CMEF and OF. The values are averages over 100 network realizations with 90% confidence intervals as error bars.

Chapter 7

Summary of the results

Table 7.1 and Figure 7.1 summarize the results from each of the used method. The Moving window algorithm with one cut (MWA) gives a considerably high upper bound for the performance since it does not consider the interference in horizontal direction. The same algorithm with two (MWA_2) or infinite number of cuts (MWA_∞) gives more moderate values. The dimensionless mean progress of a packet given by the MWA with infinite number of cuts is the maximal forwarding capacity over one time slot. The same performance cannot be achieved in consecutive time slots, and thus, it still gives an upper bound for the maximal achievable flow.

The greedy and reverse greedy methods approximate the result of MWA_∞. By greedily choosing the used links, they give a lower bound for the maximal forwarding capacity over a single time slot. The Opportunistic forwarding (OF) and Current maximum expectation forwarding (CMEF) methods represent actual forwarding methods and they thus give a distinct lower bound for the real maximal forwarding capacity. OF also demonstrates what can be achieved with random access through local coordination. To improve viable forwarding methods, one would, thus, have to consider more global coordination or more efficient access methods.

The results from the random networks are notably worse than the ones obtained from regular networks. In square and triangular lattices, the dimensionless mean progress of a packet alternates around 0.2 except for the peaks that appear at specific intervals. The first peak of the square grid is at 1/3 which is a little more than in the triangular grid. It is noteworthy that the typical level of performance in regular lattices is about the same as the maximum capacity of a random network in one time slot. This would suggest that, at best, it is possible to find, or pick out, a set of nodes that resembles a regular structure from a random network.

The hexagonal lattice forms an exception to the other studied regular structures. Because the nodes are not in symmetric positions, the dimensionless mean progress is lower with the shortest transmission radii. When the mean number of neighbors

grows, the 0.2-level is achieved also with hexagonal lattice.

Table 7.1: The maximal dimensionless mean progress, u, and the corresponding N_R for various methods discussed in this study.

	u^*	N_R^*
MWA	0.461	21.6
MWA_2	0.287	12.5
MWA_∞	0.199	9.89
Reversed	0.173	7.54
Greedy	0.167	6.98
OF	0.065	19.0
CMEF	0.047	12.0

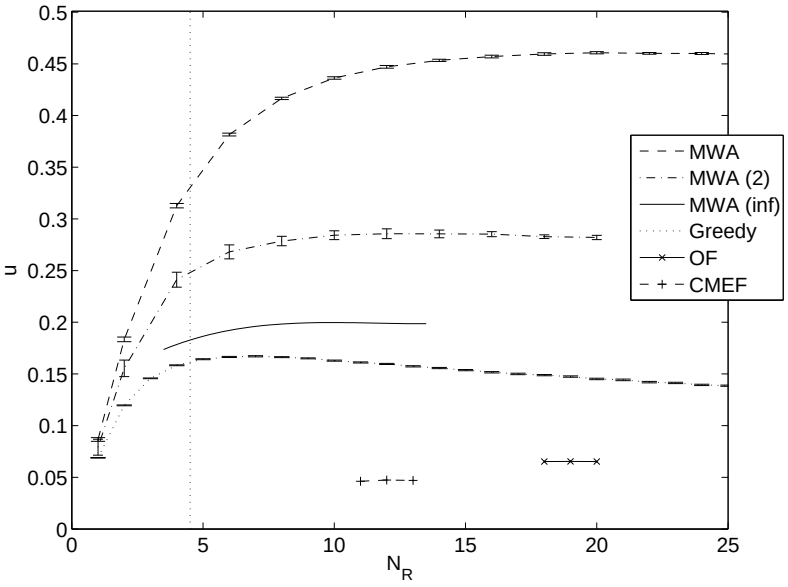

Figure 7.1: The dimensionless mean progress of a packet, u, as a function of the average number of neighbors, N_R, for various discussed methods.

Chapter 8

Conclusions

8.1 Summary

A collection of self-configuring autonomous devices acting both as terminals and routers connected with wireless links is referred to as an ad hoc network. We began this study with a brief overview on these networks, and their special case; wireless sensor networks (WSN).

Traditional ad hoc networks are most suitable for collaborative applications that require rapid deployment. So far, they have been used mainly for military purposes. The distinct features of ad hoc networks compared to other networks place several special requirements to the MAC and routing protocols. Since there is no central control, contention-free channel access techniques are difficult to implement, and contention-based MAC protocols including random access and dynamic reservation protocols are most often used. The traditional proactive and reactive routing protocols have their deficiencies when the size of the network starts to grow. Geographic routing, closest to the approach of this study, provides scalability dependent on the location service.

A high number of failure-prone, densely deployed nodes with very limited energy supplies, frequent topology changes, and many-to-one/one-to-many traffic are some of the characteristics that distinguish WSNs from normal ad hoc networks. The primary goal of a MAC protocol is to operate in an energy-conserving manner, which usually means turning the receiver off for certain periods. When it comes to routing, data-centric protocols utilize the common redundancy in the data, and hierarchical protocols provide scalability. Geographic routing protocols for WSNs are still few in number.

In a large ad hoc network, the macroscopic and microscopic levels can be separated. At the microscopic level, the local forwarding decision depends only on the direction given by the macroscopic level, and while the microscopic level handles the scale

of a single hop, the macroscopic level working on the scale of a source-destination path treats the network as a homogeneous, continuous medium. The assumptions imply that when considering a single direction at a time, there exists a maximum for the flow of packets that any MAC protocol can sustain. This maximum flow is a characteristic constant of the medium.

In the main contribution of this thesis, we defined a network model and found upper bounds for the characteristic maximum flow under that model. The model of the large ad hoc network consisted of nodes distributed according to a homogenous Poisson point process in two dimensions. The transmission range was fixed to be common for all the nodes and a Boolean interference model was used to model the wireless medium. The unbounded size of the network allowed us to simplify the model by focusing on relay traffic only. Additionally, slotted time was assumed.

With a fixed schedule the model corresponds to a flow network. According to the Max-flow min-cut theorem, the maximal value of a flow on a flow network is equal to the minimal capacity of a cut. If we examine a subset of all the cuts, we get an upper bound for the capacity by maximizing the min-cut with respect to the schedule.

For evaluating the capacity of the minimal cut, a Moving window algorithm was proposed. The algorithm limits the set of cuts to an arbitrary cut corresponding to a straight line perpendicular to the direction of the packet flow. By weakening this strict limitation of the cut set, we were able to get tighter upper bounds for the forwarding capacity. Specifically, we considered the cases with 2 straight cuts and infinite number of straight cuts.

For comparison, also actual forwarding methods and networks with a regular structure, where determining the true capacity is possible, were considered along with some approximative methods.

8.2 Further work

Although the study gave some insight into the maximal achievable forwarding capacity of the infinite homogeneous wireless network in question, the upper bound for the performance is still three times the highest achieved dimensionless mean progress. What happens in between, is an open question.

One problem with wireless networks, that distinguishes them from, e.g., wired ones, is that finding the cut with the minimal capacity is not possible without knowing the optimal schedule. Instead, one could try to locate the cut with the smallest maximal independent set of links crossing it. This is not enough for deducing the actual capacity since the schedule still affects the time share those links receive, but it would seem a reasonable basis for obtaining some kind of approximation or bounds for the quantity in question. As a whole, though, this rules out many algorithms

developed for finding the minimal cut.

Since the dual problem of finding the minimal cut is somewhat problematic, searching for the optimal schedule may provide more distinct information about the optimal ways to forward traffic. By limiting our consideration to the straight cuts in the proposed algorithms, we got results that are only achievable over one time slot. Methods that produce schedules where the corresponding flow network is connected would help to raise the lower bound of the achievable capacity. The given forwarding methods used only local coordination to spread the traffic, and methods able to create a connected network would demonstrate the benefits from more global coordination.

The interference model selected for the analysis, namely the Boolean model, is not a very accurate representation of the real world. For the results to have more significance when it comes to evaluating existing protocols, a more realistic interference model would come in handy. Also, the effect of the interference model on the optimal parameters, in addition to the performance, has some relevance to the network design.

The efficiency of one-to-one traffic in large ad hoc networks does not seem to be too high in general. This would suggest widening the area of interest into broadcast type of traffic, which is more typical in wireless sensor networks that have possible applications requiring a high number of nodes. Although the problem statement of this study is very theoretical, and the possible utilization of the global coordination required to enhance the forwarding capacity with actual MAC and routing protocols is a completely separate question, a link to the real world might be closest to the WSNs.

Appendix A

Moving window algorithm

Table A.1: Moving window algorithm

0. $i = 0$, $W = \emptyset$, $L = \emptyset$, $S_0 = 0$
1. **while** $\sum_{j=0}^{i} S_j < M$ **do**
2. $i := i + 1$
3. $S_i \sim \mathrm{Exp}(2\lambda R)$
4. **forall** $w \in W$ **do**
5. **if** $\sum_{j=0}^{i} S_j - y(w) > 3R$ **do**
6. $W := W \setminus \{w\}$
7. **forall** $l \in L$ **do**
8. **if** $t(l) = w \vee r(l) = w$ **do**
9. $L := L \setminus \{l\}$
10. TREE→REMOVE(l)
11. **end if**
12. **end for**
13. **end if**
14. **end for**
15. $x(\hat{w}) \sim \mathrm{Uni}(2R)$, $y(\hat{w}) := \sum_{k=0}^{i} S_k$
16. **if** $y(\hat{w}) < R$ **do**
17. **forall** $w \in \{v \in W \mid y(v) \geq R\}$ **do**
18. **if** $d(w, \hat{w}) \leq R$
19. $t(\hat{l}) := \hat{w}$, $r(\hat{l}) := w$
20. $L := L \cup \{\hat{l}\}$
21. TREE→ADD($\hat{l}$)
22. **end if**
23. **end for**
24. **else do**
25. **forall** $w \in \{v \in W \mid y(v) < R\}$ **do**
26. **if** $d(w, \hat{w}) \leq R$
27. $t(\hat{l}) := w$, $r(\hat{l}) := \hat{w}$
28. $L := L \cup \{\hat{l}\}$
29. TREE→ADD($\hat{l}$)
30. **end if**
31. **end for**
32. **end if**
33. $W := W \cup \{\hat{w}\}$
34. **end while**

List of Figures

List of Tables

Bibliography

[1] N. Abramson. The ALOHA system - another alternative for computer communications. In *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 281–285, 1970.

[2] J. Jubin and J.D. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1):21–32, 1987.

[3] E. Hyytiä and J. Virtamo. On load balancing in a dense wireless multihop network. In *NGI 2006, 2nd Conference on Next Generation Internet Design and Engineering*, pages 72–79, 2006.

[4] L.G. Roberts. ALOHA packet system with and without slots and capture. *SIGCOMM Computing Communication Review*, 5(2):28–42, 1975.

[5] L. Kleinrock and F.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.

[6] P. Karn. MACA - a new channel access method for packet radio. In *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, volume 2, pages 134–140, 1990.

[7] C.L. Fullmer and J.J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 262–273, 1995.

[8] S. Kumar, V.S. Raghavan, and J. Deng. Medium access control protocols for ad hoc wireless networks: a survey. *Ad Hoc Networks*, 4(3):326–358, 2006.

[9] S. Romaszko and C. Blondia. A survey of MAC protocols for ad hoc networks and ieee 802.11. In *Proceedings of the 4th National Conference MiSSI 2004*, pages 23–33, 2004.

[10] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIG-*

COMM'94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, 1994.

[11] G. Pei, M. Gerla, and T.W. Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proceedings of IEEE International Conference on Communications*, pages 70–74, 2000.

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

[13] D.B. Johnson and D.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.

[14] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[15] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1405–1413, 1997.

[16] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of IEEE International Conference on Universal Personal Communications*, pages 562–566, 1997.

[17] F. Ducatelle, G. Di Caro, and L.M. 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.

[18] I. Stojmenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

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

[20] S.M. Das, H. Pucha, and Y.C. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *Proceedings of IEEE INFOCOM*, pages 1228–1239, 2005.

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

[22] I. Stojmenovic and X. 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.

[23] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[24] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of International Conference on Mobile Computing and Networking*, pages 267–278, 2003.

[25] B. Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[26] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[27] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, 2002.

[28] E.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.

[29] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[30] W. Hu, V.N. Tran, N. Bulusu, C.T. Chou, S. Jha, and A. Taylor. The design and evaluation of a hybrid sensor network for cane-toad monitoring. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 503–508, 2005.

[31] D. M. Doolin and N. Sitar. Wireless sensors for wildfire monitoring. In *Proceedings of SPIE Symposium on Smart Structures and Materials/ NDE 2005*, pages 477–484, 2005.

[32] T. Gao, D. Greenspan, M. Welsh, R.R. Juang, and A. Alm. Vital signs monitoring and patient tracking over a wireless network. In *Proceedings of the 27th Annual International Conference of the IEEE EMBS*, pages 102–105, 2005.

[33] S. D. Glaser. Some real-world applications of wireless sensor nodes. In *Proceedings of SPIE Symposium on Smart Structures and Materials/ NDE 2004*, pages 344–355, 2004.

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

[35] G.J. Pottie and W.J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[36] A. Woo and D.E. Culler. A transmission control scheme for media access in sensor networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 221–235, 2001.

[37] B. Sundararaman, U. Buy, and A.D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[38] K. Kredo and P. Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961–994, 2007.

[39] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

[40] M.C. Vuran and I.F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(2):316–329, 2006.

[41] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling data-centric routing in wireless sensor networks. Technical report, University of Southern California, 2002.

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

[43] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185, 1999.

[44] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1):2–16, 2003.

[45] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, 2000.

[46] L. Li and J.Y. Halpern. Minimum-energy mobile wireless networks revisited. In *IEEE International Conference on Communications (ICC), June 2001.*, pages 278–283, 2001.

[47] T. Melodia, D. Pompili, and I.F. Akyildiz. On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(3):520–532, March 2005.

[48] R. Diestel. *Graph theory*. Springer, 2005.

[49] K. Xu, M. Gerla, and S. Bae. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks. In *Proceedings of IEEE Global Telecommunications Conference 2002 (GLOBECOM '02)*, pages 72–76, 2002.

[50] A. Iyer, C. Rosenberg, and A. Karnik. What is the right model for wireless channel interference? In *QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*, 2006.

[51] R. Meester and R. Roy. *Continuum Percolation*. Cambridge University Press, 1996.

[52] H. Kesten. The critical probability of bond percolation on the square lattice equals 1/2. *Communications in Mathematical Physics*, 74:41–59, 1980.

[53] M.E.J. Newman and R.M. Ziff. Efficient monte carlo algorithm and high-precision results for percolation. *Physical Review Letters*, 85(19):4104–4107, 2000.

[54] J.C. Wierman. Bond percolation on honeycomb and triangular lattices. *Advances in Applied Probability*, 13(2):298–313, 1981.

[55] M.F. Sykes and J.W. Essam. Exact critical percolation probabilities for site and bond problems in two dimensions. *Journal of Mathematical Physics*, 5:1117–1127, 1964.

[56] P.N. Suding and R.M. Ziff. Site percolation thresholds for archimedean lattices. *Physical Review E*, 60(1):275–283, 1999.

[57] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of IEEE Infocom*, pages 1079–1088, 2002.

[58] E. N. Gilbert. Random plane networks. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):533–543, 1961.

[59] J. Quintanilla, S. Torquato, and R.M. Ziff. Efficient measurement of the percolation threshold for fully penetrable discs. *Journal of Physics A*, 33(42):399–407, 2000.

[60] J.T. Virtamo. A model of reservation systems. *IEEE Transactions on Communications*, 40(1):109–118, 1992.

[61] O. Apilo, P. Lassila, and J. Virtamo. Performance of local forwarding methods for geographic routing in large ad hoc networks. In *Proceedings of The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)*, 2006.

[62] J. Nousiainen. Local forwarding methods in large ad hoc networks. Technical report, Helsinki University of Technology, 2007.

[63] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. *SIGCOMM Computer Communications Review*, 35(4):133–144, 2005.