NTNU
Norwegian University of
Science and Technology

Capacity of Mobile Ad Hoc Networks Using an Airborne Relay

Kristian Buhaug Solbakken

Master of Telematics - Communication Networks and Networked Services
Submission date: June 2016
Supervisor: Øivind Kure, ITEM
Co-supervisor: Erlend Larsen, FFI

Norwegian University of Science and Technology
Department of Telematics

Problem description

Title:	Capacity of mobile ad hoc networks using an airborne relay
Student:	Kristian Buhaug Solbakken

Increasing need for data capacity in military networks makes it interesting to utilize higher frequencies to achieve higher data rate. However, due to absorption, higher frequencies have a shorter range. Mobile ad hoc network (MANET) can create paths between neighboring nodes in a store and forward fashion, and thereby provide connectivity between hosts over longer distances even in environments where static infrastructure is impractical.

Although the use of MANET seems promising at first glance, experience from earlier work suggests that the available capacity is not that encouraging. In order to further improve the network performance, it is interesting to study how airborne nodes with a high vantage point can aid ground-based mobile networks to increase available capacity. A higher position reduces the amount of obstacles between nodes which in turn result in a better range with fewer hops between the source and destination of traffic, and might increase network capacity.

The thesis will through a simulated testbed study how airborne relay nodes flying over a ground-based MANET can affect the available capacity.

Responsible professor:	Øivind Kure, ITEM/UNIK
Supervisor:	Erlend Larsen, FFI

Abstract

Self-organizing wireless networks is an interesting technology with a potential of providing robust communication in environments without existing infrastructure and a minimum of configuration. Such features seem especially relevant for the tactical domain, with a high degree of mobility and an increasing demand for information. While ad hoc networks show promise when it comes to robustness, a lot of research point out how the capacity can be a challenge. Wireless communication is confined by the shared medium within the range of the radio and forwarding leads to fewer resources available to each station.

One proposal to address the challenges of Mobile Ad hoc Networks (MANET) is to use Unmanned Areal Vehicles (UAV) with communication capabilities. A UAV can serve as a relay placed above communicating ground stations and provide a path with fewer hops. Communication over fewer hops can lead to lower delay at the same time as the ground network is offloaded.

The Norwegian Defence Research Establishment (FFI) have an experimental lightweight communication module equipped with wireless IP interfaces. The module is able to run MANET routing protocols. This thesis aims to introduce readers to relevant technologies and challenges present in mobile ad hoc networks. Based on this knowledge, the thesis develops and measures the performance of networks utilizing a UAV as a relay.

From the measurements, this thesis finds that there are scenarios where an airborne relay can provide low latency communication. In a grid of nodes, simulation imply that by forwarding data destined to nodes with many hops through a two hop relay path lower delay can be obtained.

Sammendrag

Trådløse kommunikasjonsnettverk med mulighet for å organisere seg selv er en interessant teknologi med mulighet for å tilby robust kommunikasjon i situasjoner uten eksisterende infrastruktur og et minimum av konfigurasjon. Slik funksjonalitet fremstår spesielt relevant for det taktiske domenet. Her det er høy grad av mobilitet og et økende behov for informasjons utveksling. Selv om nettverk som organiserer seg selv fremstår som en lovende teknologi med tanke på robusthet, peker en rekke forskning på hvordan kapasitet kan være en utfordring. Trådløs kommunikasjon begrenses av et felles medium innenfor en radiostasjons rekkevidde. Samtidig fører retransmisjon til at de tilgjengelige resursene for hver radio reduseres.

Et forslag mot utfordringene i mobile selv-organiserende nett er å bruke ubemannede flyvende fartøy (UAV) med støtte for radio kommunikasjon. En UAV kan fungere som et rele plassert over kommuniserende bakkestasjoner og tilby ruting igjennom færre stasjoner ved kommunikasjon mellom enheter utenfor rekkevidde. Færre retransmisjoner fører til lavere forsinkelse samtidig som bakkenettet avlastes.

Forsvarets forskningsinstitutt (FFI) har en eksperimentell, lett kommunikasjonsmodul utstyrt med trådløs IP-grensesnitt. Modulen støtter også ruting-protokoller egnet for selvorganiserende nett. Denne oppgave søker å introdusere leseren for relevant teknologi og utfordringer som er gjeldende i mobile nettverk. Basert på denne innsikten beskriver oppgaven simulerings-modeller og utfører målinger på nettverk som benytter en UAV som kommunikasjonsrele.

Ut fra utførte målinger viser oppgaven at det finnes situasjoner hvor et rele plassert på en flyvende plattform er i stand til å gi lav forsinkelse. Simulasjon på stasjoner plassert i et rutenett indikerer at ved å videresende data som er addresert til noder med mange rutehopp over et rele, kan lavere forsinkelse oppnås.

Preface

Six months ago I started this project as the final delivery of my master thesis and the fulfillment of my Masters of Science degree in Telematics — Communication Networks and Networked Services at the Norwegian University of Science and Technology (NTNU). This document is the report of this process. All research has been executed at the department's student offices in Trondheim.

The project was undertaken at the request of the Norwegian Defence Research Establishment (FFI). My research questions are the result of discussions and input from my supervisor Dr. E. Larsen and professor Ø. Kure. At times, the research has been quite time-consuming, but with such an interesting topic, I have had a great time.

I would like to take the opportunity to thank both my supervisor and professor for introducing me to the subject and the cooperation. To my classmates, I would like to thank you for the company and the good time during my two years at NTNU. You are what makes Trondheim, Norway's most popular student town.

Kristian Buhaug Solbakken
Trondheim, Norway
June 2016

Contents

List of Figures

List of Tables

List of Acronyms

ACK Acknowledgment.

AODV Ad Hoc On-Demand Distance Vector.

API Application programming interface.

ARP Address Resolution Protocol.

CRC Cyclic Redundancy Check.

CSMA Carrier Sense Multiple Access.

CSMA/CA CSMA with Collision Avoidance.

CTS Clear to Send.

DCF Distributed Control Function.

DiffServ Differentiated Services.

DIFS DCF Interframe Space.

DSCP Differentiated Services Code Point.

DTN Delay-Tolerant Network.

FAMA Floor Acquisition Multiple Access.

FFI Norwegian Defence Research Establishment.

IEEE Institute of Electrical and Electronics Engineers.

IP Internet Protocol.

ISO International Organization for Standardization.

LAN Local Area Network.

LLC Logical Link Control.

MAC Media Access Control.

MANET Mobile Ad hoc Network.

MPDU MAC layer PDU.

MPLS Multiprotocol Label Switching.

NAV Network Allocation Vector.

OLSR Optimized Link State Routing Protocol.

OSI Open System Interconnection.

PDU Protocol Data Unit.

PLCP Physical Layer Convergence Protocol.

QoS Quality of Service.

RTS Request to Send.

SIFS Short Inter Frame Space.

SNR Signal-to-Noise Ratio.

TCP Transmission Control Protocol.

ToS Type of Service.

TTL Time To Live.

UAV Unmanned Aerial Vehicle.

UDP User Datagram Protocol.

UHF Ultra High Frequency.

VHF Very High Frequency.

YANS Yet Another Network Simulator.

Chapter 1 Introduction

Traditionally, radio communication used in the tactical domain is based on voice communication through radios in the Very High Frequency (VHF)-band. However, as the tactical level experience an increasing adoption of digital aids, the demands for a high capacity communication infrastructure rises. It is an expectation today that some of the services used in strategic and wired networks also will be available for the feet on the ground. Higher capacity at the tactical level allows information such as intelligence and progress to be shared more rapidly. Modern communication infrastructure also comes with the promise of interoperability and the possibility of network-centric warfare.

Another factor that fuels the expectation for high performing communicating infrastructure even more, is the wide adoption of high capacity cell-phone technology in the civilian world. This technology is seemingly able to deliver high data rates with small equipment at nearly no cost. Whereas, this technology requires quite an extensive infrastructure of base stations, connected by wire.

The most obvious answer to increasing capacity demands is to increase the frequency spectrum. In compliance to the Shannon-Hartley theorem [5, 6], the maximum data rate of a channel is strictly tied to the signals' bandwidth and background noise. In general, information is transferred by changes to a signal over time. Since the frequency of a signal is a literal measure of changes, the ability to transfer data can increase with higher bandwidth. By increasing the channel frequency, the usable bandwidth would also increase.

One advantage of lower frequencies though is that the signals have better penetration, meaning that they are better suited to pass through obstructions such as buildings and trees with less attenuation. When deploying a wireless network on lower frequencies, each participant is able to talk to a bigger share of the network directly. In turn, fewer repeaters or base stations need to be deployed for the network to stay connected. The bottom line is that higher frequencies can offer a better capacity at short range, but at an expense of more repeaters/base stations to stay connected. To compensate for the short range, wireless mesh networks have increased in popularity.

Devices with wireless interfaces that can be connected to each other without extra infrastructure sound alluring. On the other hand, if higher capacity is the goal, it can prove to be a challenging path.

As one of the most basic concerns, the capacity Mobile Ad hoc Network (MANET) has received much attention from research. The studies suggests that the available capacity of wireless ad hoc networks are limited. Most studies conclude that the network capacity is limited by the number of stations in the network. While a low-density network can offer spatial reuse and higher capacity, the probability of network separation increases. At the same time, it becomes clear that networks with a high density, suffer from the fact that more time is spent forwarding neighboring nodes information at the same time as the probability of collisions in random access networks increases.

Ongoing work at Norwegian Defence Research Establishment (FFI) investigates the usage of commercial network technology in combination with MANET routing protocols placed inside a lightweight unit suitable for mounting on an Unmanned Aerial Vehicle (UAV). An elevated communication relay placed aboard a UAV or another airborne platform might offer a path between nodes on the ground with less attenuation, and thus, provide a two-hop path with reduced delay. At the same time the load in the ground network is reduced. On the other hand, too much use of the relay may prove inefficient. Heavy use of the relay may cause the relay link to be saturated, leaving the relay link slower or less reliable than existing ground links. Furthermore, if the capacity is increased, how should it be shared between stations to provide fair and rational use of the resources?

One option is to divide traffic into different classes of traffic and per hop policing for how different traffic should be distributed. This thesis follows an approach where traffic intended for relay transfer can be marked. A routing protocol then enforces a per hop behavior of shortest path while forwarding marked packets via a flying relay node.

The report uses simulation models of existing and available commercial technology for wireless ad hoc networks to compare capacity found in traditional ground networks, with the capacity found in networks that utilize an airborne relay node.

Based on the problem raised by FFI, the following questions are raised:

- How does the throughput capacity of two classes of traffic scale with the number of nodes and the distribution of traffic over a relay and normal traffic?
- How does the capacity in a relay-aided network compare to that of a pure ad hoc network?

To answer the questions, the thesis starts with a short introduction to existing research and technology. Based on the background theory, the thesis models a simple routing protocol, a minimal propagation model and a traffic generating application. Together with a discrete network simulator, the models can simulate a network of ground nodes, using an elevated relay for forwarding under different network topologies. The thesis focuses on two different routing strategies. First, a network with two traffic classes is evaluated under different amounts of load. The two classes are divided between the elevated node and the terrestrial network. Next, a strategy to limit the number of intermediate nodes of long paths is simulated. Here, packets destined for a long ground path are forwarded over a shorter relay-path to allow lower delay. The focus is on how a shared medium limit capacity, routing of information and potential performance gains from shorter logical paths. With this in mind, the thesis makes arguments under the assumption that all nodes are within range of the relay and ground nodes are limited by a fixed range.

1.1 Outline

The rest of the thesis is organized as follows: Chapter 2 gives an overview of relevant theory and theoretical background. This chapter gives an unfamiliar reader a foundation and serves as the backdrop for the rest of the report. Chapter 3 will be used to describe development of a network simulator. Next, the simulated test bed is used to perform measurements on a set of scenarios chosen to illustrate aspects of relay forwarding. Chapter 4 combines simulation results and assess their potential implication. The thesis concludes with a discussion of the presented results validity and relevance in Chapter 5, before a concluding summary of findings in Chapter 6 ends the report.

Chapter 2
Background

The following chapter starts with a short introduction to a conceptual communication model. The model partitions communication systems in logic layers and is later used when introducing relevant theory. This background information creates the foundation for the rest of the report. The second section gives an introduction to network routing. This is to give an understanding of how information is forwarded in computer networks. Next, the report describes characteristics of commercial wireless networks and a introduction to its performance. The combined understanding of routing and the constraints of wireless communication in MANET is then used in an overview capacity in MANET. This last section uses prior research in information theory to describe the theoretically available capacity and how it is measured.

2.1 A Layered Interconnection Model

Given the complexity of modern communication infrastructure, it is beneficial to organize functionality and characteristics into a structured framework. Fortunately, a standard for the communication functions of networking systems already exists and is well adopted. This thesis uses the Internet protocol stack, a simplification of the International Organization for Standardization (ISO) Open System Interconnection (OSI) model. [7] The model separates Internet protocols into five layers illustrated in 2.1. At each layer devices can exchange data in interfaces known as layer N-protocols. [1]

In the same way as the OSI stack, data communication between two devices is formed on the topmost layer (Application layer). A Protocol Data Unit (PDU) carries the information between layers. When transferred to the layer below, the PDU becomes this layer's payload. The PDU is concatenated with a header and sometimes a footer, producing a new level-N PDU. This process continues until layer 1 (Physical). Here, data is shifted to another device. At the receiving device, the whole process is reversed. Each layer removes its header and hands the PDU to the higher layer.

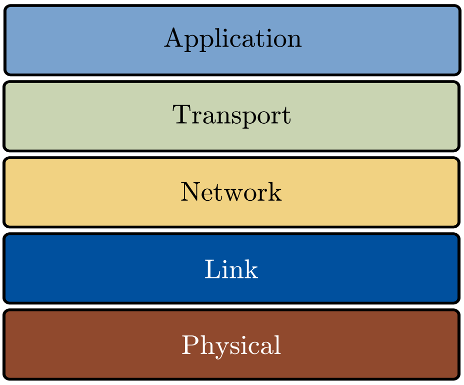


Figure 2.1: The five-layer Internet stack [1]

Finally, the topmost layer receives and consumes the original payload.

1. The physical layer is responsible for the physical movement of data between devices. It defines the electrical specifications and the relationship between a device and a transmission medium, such as voltage, timing, and frequency.

2. The data link layer provides a logic link between directly connected nodes and detects faults that may occur at the physical layer. Here, flow control between devices is defined and the establishment and termination of a connection are handled.

3. The network layer implements the moving of packets across intermediate routers between two endpoints. There is no guarantee of message delivery, but error detection is normal.

4. The transport layer gives the function of transfer data over one or more networks. Some protocols also provide reliable messaging.

5. The application layer interacts directly with the software application creating and consuming the data.

2.2 Network Routing

When a collection of hosts are connected together in a network; routing is the process of selecting a forwarding path from one node in the network to the next until delivery at the destination. Routing is performed in different kinds of networks, including circuit switched networks such as telephone networks and packet switched networks like the Internet. For networks connected with a cable, this is usually done by receiving on one interface and sending the packet out on another. In packet switched

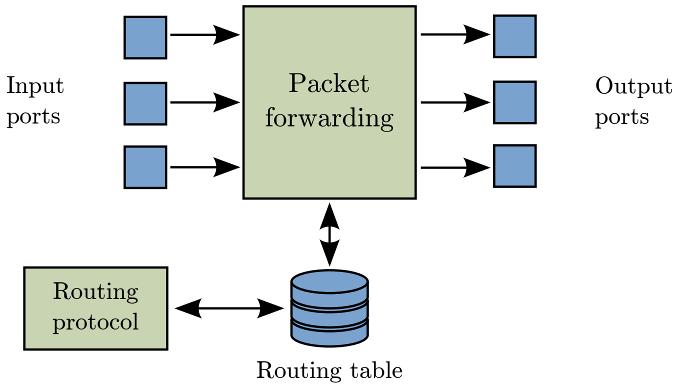


Figure 2.2: Generic network router architecture

networks information is sent from host to host as small chunks of information and handled individually at the arrival of each intermediate hosts. Circuit switched network routing start with an initial setup where links between the source and destination are reserved for the duration of the session.

In packet-switched networks forwarding of packets are usually based on a routing table stored at each routing entity. The entity maintains a record of where information should be forwarded based on the packet's destination. Construction and maintenance of routing tables are very important for the routing to be efficient. The router query the routing table with the destination address found in incoming packets. If a record is found, it tells the router the correct output interface and the address of the next hop.

Figure 2.2 illustrates a general overview of how the ports with processing and buffers cooperate with the routing table and a routing protocol. Including the interfaces, a general router node participating in a network consist of one or more processing modules, buffering modules, and an internal connection (switching fabric). With packets arriving at the interfaces, the aggregated data rate needs to be processed, buffered and relayed onto the correct interface. To speed up packet handling, the processing, and buffering modules may be replicated at each interface to allow for concurrent operation.

For traditional routers, each block found in the figure can represent specialized hardware. However with reduced cost and increasing computing capacity, it is not uncommon today that a router is implemented in software. In software routers, packet forwarding and the routing protocol run on a common processor. Although virtual routers implemented in software are not expected to perform like dedicated hardware, the flexibility and price can talk in its favor. Even large content providers deploy software defined networking.[8] Nevertheless, the components described are useful when studying the functionality of a router. The same is true when implementing a simulation model of a router. The components divide the operation into manageable parts.

Most routing protocols do not hold more than one record for each destination at a time. There are however protocols that support multipath routing. A routing protocol which stores multiple alternative paths can be useful when more than one path is available and the paths have different characteristics. There might be reasons for some types of traffic to always take another path than the rest of traffic. This can come out of different demands for delay or throughput, but can also be that some paths are considered more reliable or secure. Another reason for multipath routing is to spread the load over multiple paths. In situations with more than one available route and the goal of the implementation is to implement some kind of Quality of Service (QoS), the protocols usually decide which route to forward packets by a metric calculated by the protocol or a configured priority.[9] The proposed routing protocol in this paper is a protocol with potentially two routes to each destination. The routing table contains routes only between ground nodes and routes that take the relay into account using a shortest-path algorithm.

2.2.1 Dynamic Routing Protocols

In small networks with few changes in topology the routing tables may be configured statically. Larger networks with complex topologies that can change, make the manual configuration of routing tables unfeasible. Dynamic routing tries to solve this problem by creating the tables automatically based on information collected and carried by routing protocols, allowing the network to act nearly autonomously. In MANETs, the need for a dynamic routing protocol is even more evident. Mobile stations results in high demands when it comes to maintaining valid routes.

A dynamic routing algorithm adapts the routing paths the traffic load or topology changes. Updates can be run either regularly or in response to a change in configuration or the topology. Even though adaptive routing protocols are more sensitive to link or topology changes, they are also exposed to problems like routing loops and oscillation.[1]

Routing protocols can also be divided into a classification of active or reactive protocols. Reactive routing protocols perform updates to routes when needed, whereas active routing protocols updates routes continually to minimize delay. Ad Hoc On-Demand Distance Vector (AODV) [10] and Optimized Link State Routing Protocol (OLSR) [11] is two of the accepted routing protocols commonly found in MANETs

AODV

AODV is a reactive protocol. If a node seeks to communicate to a node in the network which it has no route, the protocol will try to establish a route. Initialization of a request is performed by transmitting a route request that is flooded through the other nodes. When the route request reaches either the destination or a node with a valid route to the destination it answers with a route reply. Nodes monitor the status of their next hop neighbors. When a connection to a neighbor with an active route breaks, a route error message is used to notify other nodes of the loss of the link.

OLSR

A proactive protocol like OLSR seeks to maintain a continuously updated understanding over the topology. In theory, the whole network should be known to all nodes. Compared to the AODV, this results in a constant overhead of routing traffic, but with less initial delay in communication caused by route requests.

Like the name implies, OLSR uses link-states in an optimized manner. Conventional link-state protocols, flood the network with link information. OLSR uses a similar approach, but because the protocol is specially developed for wireless networks, the flooding is optimized to save bandwidth. To maintain a complete understanding of the topology, OLSR regularly transmits hello and topology control messages. Hello messages are used to detect one-hop nodes and their direct neighbors. A distributed election chooses a set of multipoint relays among the respondents of the hello. These multipoint relays then source and forwards topology control messages. The multipoint relay makes OLSR unique from other link-state protocols. Only a subset of nodes creates and transfer link-state and only the interfaces used in the election are advertised.

2.3 Mobile Ad Hoc Networks

In a lot of situations, it is most practical to be connected over a wireless connection — such as when connected entities need to be mobile. Nodes connected by a radio, may move freely as long as they are within the range of the network interface and with the adoption of smartphones, most people have become quite accustomed to staying connected while being mobile. Meanwhile, much of the required infrastructure to facilitate phone networks are hidden from the average user.

In many of the wireless networks seen today, there is a requirement for existing infrastructure to give authentication and configuration of moving nodes. Situations such as emergency response and military deployment to areas without existing infrastructure, cannot rely on such infrastructure to be available when needed. The grail in such situations seems to be a continuously, self-configuring network, able to manage the connection of multiple endpoints, without prior configuration and fixed infrastructure. MANET are networks of moving nodes that fit the specified requirements. In order for a network to be ad hoc and still be able to communicate past each node's limited range, nodes have to be willing to forward data on behalf of each other. When routing data there are primarily two different routing strategies.[4]

Broadcast routing retransmits all messages received, creating a flood of transmissions throughout the network.

Point-to-point forwarding limits the needed retransmissions by creating routing tables as described in Section 2.2. The routing table has to be maintained by an elected station with global knowledge of the network topology, or as decisions done by intermediate nodes.

When imagining all nodes broadcasting data, it is clear that this ensures a robust network with respect to topology changes. However, for scenarios with capacity demands, this is not a winning strategy. Most of the nodes will receive the same information from more than one repeating router.[4] In point-to-point forwarding, the routing decisions are made locally in a hop-by-hop manner based on the nodes understanding of the topology. This reduces the occupied area to a minimum.

A challenge with the point-to-point strategy, however, is the need for an updated understanding of the topology. With mobile nodes, the perception of the topology has a limited lifetime. In fact, with a changing topology, information needs to be updated more regularly to stay valid. In turn, mobility can in some cases reduce the available capacity for the actual data. The challenge seems to be to maintain a valid understanding of the network while using a minimum of the available resources.[12]

2.3.1 Multiple Access in a Shared Medium

A shortcoming in all wireless communication is the fact that everyone within an area shares the communication medium with each other. Just like in a room full of talking people; concurrent conversations are perceived as noise and make it harder to maintain a conversation. In order to ensure high throughput and provide fair access, it is wise to implement a protocol for access to the medium. This way conversation is more likely to be free of annoying interference. Generally, we can split this types of access protocols into three different categories: Random access, fixed assignment, and on-demand assignment. Since this thesis focuses on communication in ad hoc networks, fixed channel assignment is not considered relevant. For ad hoc networks to be effective, it is important to avoid fixed solutions to be able to adjust to multiple situations.

Random Access

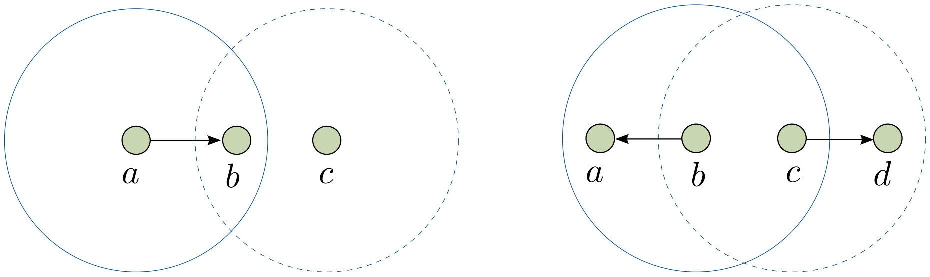
In Random access, anyone can use or reserve a channel at their will, however just as in social settings; it is considered impolite to disrupt an ongoing conversation.

An example of a protocol based on random access is the ALOHA protocol. [13] In ALOHA, stations that intend to talk simply indicate this by transmitting a hello message followed by the actual data. The hello informs neighboring stations that the channel is busy. Messages received by the recipient are confirmed with an Acknowledgment (ACK). Unacknowledged messages are considered lost and will be retransmitted after a set timeout period. The random nature of ALOHA results in high rates of collisions and poor performance. Keep in mind that the start of messages can come at any time. Thus, a frame can be rendered useless even when a collision happens right at the end of a transmission. The result is that both transmitters need to wait and try again. Slotted ALOHA improves on this by declaring time slots where a frame is allowed to start. This improves capacity by ensuring that all collisions happen in the beginning of a message.

Another common protocol used in random assignments is Carrier Sense Multiple Access (CSMA). In a CSMA-network, participants try to avoid a collision by verifying that a medium is free prior to transmitting. If a channel is idle a node might send its queued message directly, with a probability p or after a random wait. Later, in Section 2.5 it is presented how non-persistent CSMA is used in commercial wireless interfaces. Wireless communication cannot directly detect collisions and always transfer messages in their entirety. To avoid collisions wireless networks usually improves the performance of CSMA by implementing CSMA with Collision Avoidance (CSMA/CA).

By reserving a channel with a hello message, it is possible to tell nearby stations that the channel is busy and that they should not disturb. However, there is still a chance that nodes outside the range of the transmitter interfere with the receiving node. The challenge is referred to as the hidden node problem.[14, Ch. 4] In Figure 2.3a on the next page it is clear how this play out. Even though node a reserves b, there is no way c can know that this has taken place, and c are not able to tell that a conversation is going on. The result is that c might disturb the conversation between a and b even though a, reserved the medium and c checked if the medium was clear.

For the hidden node problem to arise, b and c need to be within range. b can inform c that it is reserved to a, but not while receiving. With a handshake protocol where a sends a Request to Send (RTS) to b who answer with a Clear to Send (CTS) prior to using the channel, c can deduce that b is occupied with a. When b ACK the frame from a, c is informed that b is free again. CTS/CTS solves the challenge with the hidden node problem. At the same time, it can solve the problem known as the



(a) The hidden node problem. Station b can experience interference from c when transmitting to a because a has no way of knowing if c is transmitting

(b) The exposed node problem. Node b cannot transmit to a at the same time as c transmits to d, even though they will not interfere

Figure 2.3: Problems in a shared medium partly solved by RTS/CTS

exposed node problem. Figure 2.3b illustrate another phenomenon that appears in random networks. Even though node a and d do not interfere with each other. b and c will be prevented from communicating simultaneously. Since c will overhear the RTS sent from b, but not the CTS, c can assume that a is out of range. However, if the protocol relies on reception of an ACK to confirm correct reception and frames are of variable length, it might not be safe for c to communicate with d. c will interfere with b.

On-Demand Assignment

On-demand assignment tries to combine the best of random and fixed assignments. Here the assignment can be adjusted according to the need at the same time as the resources used to manage access can be reduced. Floor Acquisition Multiple Access (FAMA) [15] uses the handshake protocol presented in the previous section to reserve a channel (floor) before data is transferred over it. However here the handshake is placed in a fixed separate channel from where payload communication takes place. The result is that information transferred over channels other than the agreement channel can be transferred without interference. The use of multiple channels in a network can extend the available capacity beyond capacity given by spatial reuse by providing more concurrent transmissions.[14]

An advantage with the combination of multiple channels and CSMA is that it opens for a simple implementation of QoS. Having multiple channels, network capacity can be differentiated to a number of channels for a specified time. A multichannel approach can be used in CSMA as well. By having multiple channels and listening to each of them to decide if it is available. The challenge is for all nodes to monitor the current state of all channels at the same time.

2.4 Delay, Loss, and Throughput

Ideally, MANETs should move as much information as possible between any pairs of nodes super-fast and without any loss. Meanwhile, like most computer networks, this is not achievable. Instead, networks suffer from limited throughput, packet loss and delay.

Recall that in packet routing, information passes through a series of routers to the final destination. As packets travel from one node to the next along such a path, the information suffers multiple types of delay and changes in capacity. The sum of the nodal performance through intermediate nodes results in the final end-to-end performance. With multiple fundamental factors affecting the performance at each hop, the entire system quickly becomes quite complex. Introduce wireless links and moving nodes and it really becomes a challenge to fully understand and anticipate the performance. Network performance is an ideal candidate for simulation modeling. In simulations, it is possible to model the most important factors in separate steps at a level where their behavior still is comprehensive. Later, when the models are tested and understood, it is possible to connect them together and create complex systems. Modeling of each subsystem still requires technical knowledge. The next sections give an introduction and form the basis for this knowledge.

2.4.1 Delay

When a packet arrives at a router, the first task is to read the header information. The time it takes to read and look up the next hop is part of the processing delay. Processing does also include other tasks such as calculation of bit errors and update of the Time To Live (TTL) prior to placing the packet in the output queue.

At the queue, packets experience a delay as they wait to be transferred out on the interface. How long a packet stays in queue depends on the number of packets ahead in line. In an empty queue, newly arriving packets are transferred out the interface immediately. On the other hand, assuming that packets are handled in a first-come-first-served manner, a new packet that arrives at a queue with a lot of packets already waiting, have to wait for up to several milliseconds.

Router interfaces specify a data rate. Together with the size of a packet, the data rate denotes the time it takes to move a packet from the queue to the medium known as the transmission delay. For instance, for a 1 Mb/s interface, it takes $\frac{512\times 8}{10^6}$ seconds to transfer 512 bytes out on the interface.

Even once a bit from a packet is moved onto the radio spectrum, it needs to propagate the physical distance between the sender and the receiver. The time it takes to move a bit from one node to the next is referred to as the propagation delay. Electrical

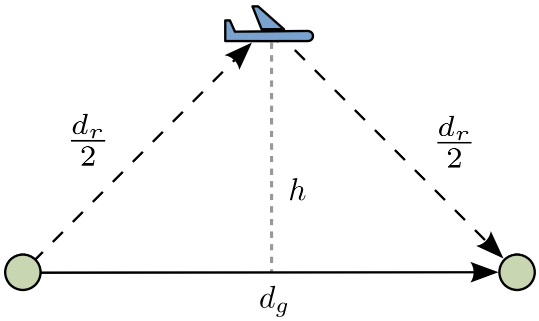


Figure 2.4: The relationship between relay and ground propagation paths

signals travel quite fast (up to the speed of light), but dependent on the medium and the distance, the propagation can make up a significant portion of the total transfer time. For long ranges such as in satellite communication, the delay is in the order of hundreds of milliseconds. For the distances typically found in wireless networks, like in this thesis, delays of more than a few microseconds are rare.

Nonetheless, it might be worth noting that the proposed shortcut over a flying relay, if seen in a physical sense is a detour. For short distances compared to the height of a relay the detour can lead to an increased propagation path according to $d_r = \sqrt{4h^2 + d_g^2}$ in Figure 2.4.

Meanwhile, for distances of less than 1 km and a theoretical propagation speed equal to the speed of light; the propagation delay stays below 3.3 µs. On the other hand, this shows how an airborne node in low altitude can outperform a satellite in orbit when it comes to delay.

2.4.2 Packet Loss

The discussion about queuing above assume an infinite queue, where the delay is the only consequence. In reality, routers do not have infinite buffers, and can only store so much information. With no place to store incoming packets, the network layer has no other choice than to remove a packet. Consequently, routers with a first-in-first-out strategy will drop new packets arriving at a full buffer. Higher layer protocols might detect lost packets and retransmit, but there is not anything the router can do about it. In Kendall's notation, this is an M/M/1/K queuing model and the time spent in the queue, or the number of packets is well understood in traffic theory.[16, 17]

Like mentioned in the section on the physical properties of MANET above, it is also possible to lose packets due to interference and weak signals. The Media Access

Control (MAC) protocol try to recover from a loss by doing a retransmission without informing upper layers. Still, this retransmission takes time.(Increase delay) Not only from the node sending but all neighboring node on the same channel. Hence, the MAC protocol Should not retransmit indefinitely. Section 2.5 will also describe how a link layer protocol lowers the sending rate when packets are lost to reduce contention. Effectively, performance does not only depend on a network's delay, but also in the terms of probability of packet loss.

2.4.3 Throughput

Now, with loss and delay covered, throughput is the last of common performance measures. Throughput is a measure of how fast a network can move information. Typically, measuring of the maximum throughput in a communication infrastructure is performed by transferring a block of data from one end to the other and measure the time needed. Clearly, the throughput cannot transfer faster than the lowest link data rate in a series of routers. If the first link in a series supports a higher data rate than the second, the first router will continue to transfer at the rate specified, meanwhile the second router will forward in its maximal next hop link rate. If this continues, the backlog left at the second router will lead to a growing queue until packets are dropped.

For some applications, such as real-time voice communication, it is preferable with a low delay and an average throughput above some level. While, a file transfer application, will accept a high delay and prefer the highest possible throughput. Delay-Tolerant Networks (DTNs) is a network architecture that fit the later requirements. In wireless networks without continuous connectivity, increased buffer space can provide reasonable throughput. Grossglauser and Tse observed in [18] that the mobility of a MANET can actually increase the average throughput. Another approach especially relevant for this thesis is [19]. Here, the author uses a UAV-relay in a load-carry-and-deliver Paradigm to ferry information between nodes. A UAV loiters over the source and fill up a packet buffer before it flies to the destination and delivers the data. Throughput results show that the load-carry-and-deliver method perform better than a chain of seven nodes doing traditional forwarding.

2.5 IEEE 802.11 Wireless Networks

Despite the development of many wireless network protocols, one group of protocol standards has evolved into being the de facto in Local Area Networks (LANs). The Institute of Electrical and Electronics Engineers's (IEEE) standard 802.11, also known as WiFi, was first released in 1997[20]. The standards committee has defined the 802-related protocols into two separate layers. Together, the Logical Link Control

(LLC) and the WiFi MAC-layer give the specification for a physical and link layer, suitable for MANET deployment.

The simulations and the following results presented later uses 802.11 as an example of a modern and available wireless protocol. WiFi is chosen primarily because of its widespread use and attention in existing research. It is also worthwhile to mention that the experimental router, developed at FFI is equipped with WiFi-interfaces. Since 802.11 already is a popular protocol it is possible to use existing simulation models which reduce the workload needed to perform simulations. Existing modules used in prior surveys also enables results to be seen in the context of other published work, which increases credibility.

Under 802.11, there are several amendments in use. One of them is the 802.11b, which this thesis use. Here, CSMA/CA in the Ultra High Frequency (UHF) band give random access to one shared channel. There is also support for RTS/CTS and an ad hoc mode for extending and building the network without central control. This is an important feature of a MANET. The ad hoc mode enables hosts to connect with each other and form a functional and autonomous link layer.

The ad hoc behavior of WiFi is described by the Distributed Control Function (DCF). To avoid interference, DCF requires hosts to listen to the shared channel for a period of time to ensure that the channel is not used by somebody else. When the channel turns idle, all hosts in the network are expected to wait for a random backoff period prior to sending. The random period is implemented to avoid multiple transmitters from perceiving the channel as idle at the same time and interfere with each other. Correct reception of a frame is confirmed by an ACK from the recipient after one Short Inter Frame Space (SIFS) period.

In more detail; the DCF specify two strategies for information transmission. The first option is a two-way handshake where each node can transmit packets immediately if the channel has been idle for a period — an ACK from the receiver confirms successful reception. If a station receives a frame before an idle period is over, the channel persists to monitor the channel until the link is found idle for a complete DCF Interframe Space (DIFS).

To minimize the probability of collision after one DIFS, each station calculates a random backoff period t_{cont}, and wait until the backoff time has passed before starting a transmission. The backoff is found by drawing a uniformly distributed integer from a range referred to as the contention window. The minimal value of the contention window is in the range of [0, 31]. A station waits for the drawn number of fixed time slots. If another station transmits before the end of a backoff period, the idle station holds the current value and continue the countdown of the backoff when the medium later appears idle. Frames that do not receive an ACK is considered lost.

Most likely, frames are lost due to collision with competing stations. Transmitting station doubles the contention window to create an exponential backoff, and thereby reduce the contention. After a successful transmission, the contention window is returned back to normal. Additionally, transmitting stations have to wait one backoff period between each following packet to avoid that one station hijacks the channel. To ensure that no station perceives a channel as idle in the SIFS between a packet transfer and acknowledgment; it is important that all stations have received the ACK before one DIFS. This is the reason that the SIFS is shorter than the DIFS.

The next alternative strategy uses a four-way handshake. A station willing to send a packet will first transmit a short RTS. Included in the RTS is the source and destination address, and the duration of the following packet transfer.(Complete with ACK). If the channel is free, the destination station will respond with a CTS. The response will contain the same duration as the request and the address of the receiver. All stations within range who receives either the RTS and/or the CTS will set their Network Allocation Vector (NAV) for the duration set in the exchanged messages. The assumption is that all stations in a network operate with fairly equal range. A stations with a more powerful transmitter than other stations can still interfere outside the allocated area.

An ACK finishes the handshake and frees the channel to other users. Because the alternative strategy uses more of the channel to set up a connection it is said to add overhead. At the same time, since RTS and CTS frames are shorter, it reduces the probability of collisions.

2.5.1 Performance

The performance of a wireless network can be measured by its throughput measured by the amount of data that can be transferred per second, the delay between sending and receiving, or the likelihood of lost packets. For a wireless link, the performance depends mainly on two factors — the bit rate of the channel and the overhead introduced by the protocol.[2] The bit rate limits how fast bits are modulated into the radio spectrum and is referred to as transmission delay. In following simulations and discussions a link with a constant rate of 1 Mb/s is assumed. The time it takes to transmit information is found by dividing the total amount of data on this data rate. On the other hand, remember that the data created in the topmost layer is packed with more data on its way down the protocol stack which introduces overhead.

The overhead introduced at the link layer also includes the time it takes to get access to the channel and the extra information sent by the DCF to control the channel allocation. The extra information increases the time it takes to transfer data at the given the data rate. Previous sections have already introduced the DIFS and SIFS

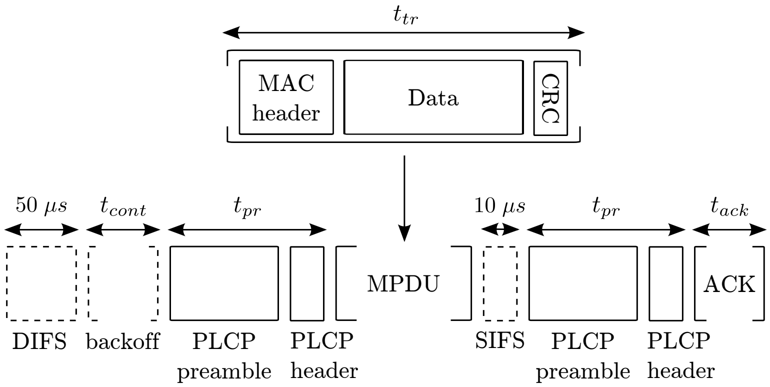


Figure 2.5: Timing in successful transmissions of a 802.11 frame without a RTS/CTS handshake [2]

intervals. Figure 2.5 takes these intervals and present them together with the rest of the time slots that add up to a successful transmission.

When a station is ready to send, the first bits that are transferred is the Physical Layer Convergence Protocol (PLCP) preamble and header. This is basically bits used by 802.11 to sync the transmitter and receiver clocks and need to precede every frame. The MAC header is 30 bytes long and provides link layer information such as addressing. Data from the network layer is transferred with a Cyclic Redundancy Check (CRC) to detect bit errors. If the bits propagated successfully to the intended recipient, the receiver waits for one SIFS and sends an ACK of 14 bytes together with PLCP information.

Table 2.1 present parameters defined in IEEE Std 802.11-2012 [21] for 802.11b. Later, this is the also the parameters used in the simulation environment. The standard defines a SIFS and time slot which is used to calculate DIFS = SIFS + 2×slot. PLCP preamble and header have a fixed length of 192 bits always transferred at a rate of 1 Mb/s. An alternative short preamble that ramps the rate of the header to a higher bit rate and provides shorter t_{pr} is described in the standard [21, Section 17.2.2.3], but for interfaces operating on 1 Mb/s this increase in rate is not feasible.

Table 2.1: Timing parameters of 1 Mb/s 802.11b in µs

SIFS	slot	DIFS	t_{pr}	t_{ack},	$\bar{t}_{cont}$
10	20	50	192	14	320

Frames transferred with a RTS/CTS handshake add even more overhead. Like Figure 2.5, the transmitter starts with one DIFS period, followed by a backoff. Next, the transmitter transfers a 20 bytes long RTS frame with PLCP preamble and header. After one SIFS the recipient answers with a CTS frame of 14 bytes, also prepended by a preamble and a header. A second SIFS period is waited before the initializing node can continue the normal transfer with a MAC layer PDU (MPDU) followed by an ACK from the recipient — both with a preamble and a header. With all parts of one packet-transfer presented; it is now possible to deduce a measure for channel efficiency. The efficiency E for frames without RTS/CTS and E' with, can be expressed by the time used to transfer data in comparison to the time it would take to only transfer the payload t_d [2]. Remember that stations that sense a channel as busy also wait for a random number of time slots. For the minimum contention window, the mean backoff time is equal to 16 times the slot time.

$$E = \frac{t_d}{\mathit{DIFS} + t_{cont} + 2t_{pr} + t_{tr} + \mathit{SIFS} + t_{ack}} \tag{2.1}$$

$$E' = \frac{t_d}{\mathit{DIFS} + t_{cont} + 4t_{pr} + \mathit{RTS} + \mathit{CTS} + t_{tr} + \mathit{SIFS} + t_{ack}} \tag{2.2}$$

In Table 2.2 the expected upper bound efficiency for transmissions with different sized packets is calculated. Because the overhead stays the same, bigger packets have a higher efficiency than small ones. Naturally, the four-way handshake does also have an impact on the efficiency.

Table 2.2: Upper bound efficiency of 802.11b at 1 Mb/s

Data bytes	t_d, ms	t_{tr}, ms	E	E'
1500	12.00	12.27	0.91	0.87
512	4.10	4.37	0.77	0.69
64	0.51	0.78	0.31	0.22

The access method in 802.11 have both advantages and drawbacks. A random access method means that the channel allocation can be distributed and that all stations have roughly the same probability of access. Thus, the access is fairly shared between all stations. However, networks with many contending stations experience a significant collision rate. As a consequence dense network can suffer unfair access. Stations who collide increase their contention window. Hence, the probability of access to the network is reduced for colliding nodes. [2]

2.6 Information Theory of MANET Capacity

Current research generally follows either of two different tracks when capacity is evaluated. One approach is to look at a theoretical track, which mainly focus on the theoretical bounds of the capacity. The other track emphasizes more on the practical aspects. By concentrating on the creation of optimal scheduling mechanisms to achieve the theoretical capacity bounds. At the same time, capacity might not be the only purpose of practical optimization. This section will present an overview of prior work along the theoretical track in order to establish a baseline for later experimental results.

When trying to find a clear definition of network capacity, it becomes clear that there is no single definition. As briefly touched upon in Chapter 1, Shannon defines channel capacity based on channel bandwidth and the signal to noise ratio [6]. The channel capacity is seen as the ratio between information put into a channel and the information retrieved at the other end. This definition is limited to a single source-destination pair and does not touch upon the consequences of repeating information to extend range.

Throughput capacity can be a measure of networks combined capacity. When the throughput is the amount of data correctly transmitted from all sources during a unit of time. In [22], the authors present throughput capacity as "the time average of the number of bits per second that can be transmitted by every node to its destination". The per node average throughput λ is then found as the arithmetic average of the collective throughputs.

$$\lambda = \frac{1}{n}\sum_{i=1}^{n}\lambda_i \tag{2.3}$$

In order to have a valid measure of network capacity, the distance between nodes can be important, especially for wireless networks where interference from simultaneous transmissions affects the result. Transport capacity presented in [22] is a popular definition taking distance into account. Transport capacity is defined as the number of bits transferred closer towards its destination or the network throughput capacity multiplied by the average distance between the source and destination nodes in the network. Note that for networks with an average distance between sources and destinations of one, the throughput capacity and the transport capacity is the same.

With respect to transport capacity, it is expected that the total capacity increases with the distribution of nodes, where the distribution must be seen relative to the transmission power and sensitivity in the transceivers in each node.[18]

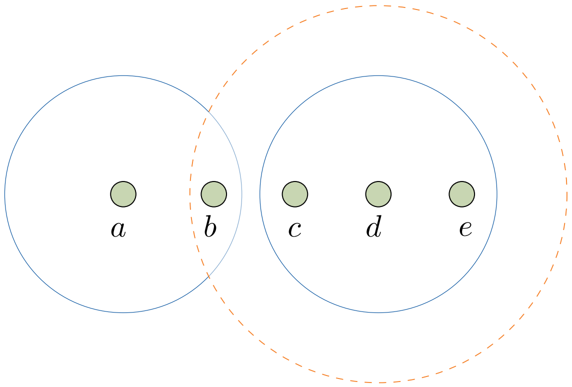
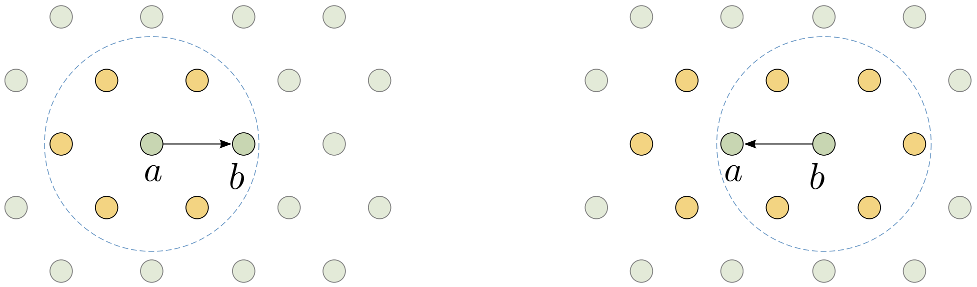


Figure 2.6: The protocol model with interference along a chain of nodes. The only nodes able to communicate at the same time is a and e [3]

When modeling interference in MANETs there are two common approaches proposed by [22], known as the physical model and the protocol model. The physical model is commonly employed when realism is important. Here technology from the physical layer such as modulation, channel coding, propagation, and Signal-to-Noise Ratio (SNR) are included in the analysis of successful reception.

Radio propagation modeling reflects the amount of energy lost between transceivers, or the received power at the receiving end by considering factors like absorption and free space path loss based on the distance between stations and attenuation compared to the transmitted power. The SNR model describes the noise and interference at the receiver and uses it to determine if reception is possible, or at what data rate reception is possible. For the protocol model, communication is considered as a binary function of the distance between all transmitters and the receiver of a message. If the condition is met, the transmission is successful, and the recipient receives at a specific rate. Otherwise, no information is transferred.

To illustrate the protocol model, think of the path taken from a source node to a destination as a series of nodes where each interferes with nearby stations along its path like in Figure 2.6. In an optimal case, each station is placed such that they only overlap with its two nearest neighbors along the path. The solid circle indicates the area of which nodes are able to receive information and the dashed circle symbolize the area where information cannot be correctly received, but interference takes place. If station a transmits a message towards e, b is not able to communicate with c before a has stopped its transfer. At the same time, c cannot communicate while a sends to b. Taking the dashed area around d into account it is apparent that the capacity is even more limited. Even though station a is outside the communication range of d, transmissions from d to e interferes with b's reception. In the illustration



(a) Neighbors of a are silenced by RTS

(b) Neighbors of b are silenced by CTS

Figure 2.7: Blocked nodes caused by reservation from central nodes [4]

showing an optimal placement of nodes along a path, only one out of four links can be in use at the time.[3]

In a situation where all nodes are placed in a regular grid with stations align both horizontal and vertical, and where stations are within range of only its four nearest neighbors, and no one else. Now, consider a scenario with only parallel traffic with all flows traveling in the same direction. Here, it is safe to assume that every third row can communicate freely. Thus, the maximum expected capacity is limited to a third of the capacity found in a straight path as illustrated in Figure 2.6. If however traffic is allowed to flow in both horizontal and vertical direction, but still originates and are destined at the ends of the grid, there exist a scheduling scheme where horizontal and vertical traffic can coexist. If access to the medium is split equally between horizontal and vertical traffic, each end node would receive half of the capacity in the parallel scenario. At the same time, for a square network, the number of sources would be doubled. Thus, the maximum total network capacity would be unchanged.

Typically studies of capacity fall into one of two different network models; arbitrary and random networks. In an arbitrary network, stations have a high degree of freedom to position themselves and choose transmission power for optimal capacity. For random networks, all nodes are randomly distributed after a uniform distribution, and nodes use a common transmission power. As a result, random distribution is expected to result in a less effective use of the area. Some places can experience no nodes and other might have a high density. With traffic destinations, there is also a higher probability for traffic to flow through the center than the edges, making the center of the network a bottleneck [3].

By taking the situation into a two-dimensional world where nodes are placed on a plane it also becomes apparent that limiting the transmission power is of high importance, illustrated in Figure 2.7. By reducing the power of each a transmission the blocked area is limited and spatial reuse is increased. If the protocol model is seen purely geometric it becomes clear that by reducing the range (radius) of a circular area blocked by a factor of two the actual blocked area is reduced by a factor of four [4]. ($A = \pi r^2$) The same argument can be made for the likelihood that a node within a distance r is chosen as the destination. With a uniform density of nodes, the probability increases with the area covered. It is also possible to use the geometric argument to state that the average distance between nodes in a source-destination pair is in the order of the square root of the area [23].

Chapter 3
Simulation and Modeling

In order to gain a better understanding and answer the problems described in the Introduction of this paper, a simulated test bed is developed. Simulation gives a practical and flexible way of analyzing systems. Compared to a mathematical model, simulation allows more fine-grained control over the granularity of a model. Mathematical models often need to be simplified to be solvable, and an implementation of the real system could clearly not be simplified at all. For a good simulation model, it is important to remove unnecessary details, while at the same time describe the relevant details. The scope of a simulation relies on its objectives.

Splitting a simulation into modules can reduce complexity [24]. A modular approach allows a study and specification of separate parts of to be done independently of the complete system. In turn, this approach may establish a higher credibility to the results gained from the experiments. When using simulation for research there are several factors that affect the trustworthiness. One important factor is repeatability and clearly defined scenarios. The following chapter gives an overview of development and details of the simulated environment.

3.1 Simulation Environment

The foundation of the simulation is created with the Ns-3 framework [25]. Ns-3 is the third in a series of network simulators and provide a C++ framework for writing discrete event driven simulators for research and education on computer networks. The advantage with a simulation environment written in a general purpose programming language supported by simulation libraries is that there are few limitations on what can be done. At the same time, ready-made building blocks reduce the effort of simulation writing. Ns-3 provide realistic models for wireless IP network technologies, as well as routing protocols and mobility management, which makes it an excellent base for further experiments.

Furthermore, Ns-3 comes with a solid random number generator based on MRG32k3a. This, 32-bit number generator is able to provide 1.8×10^{19} independent streams and a period of 3.1×10^{57}. [26] In a typical use case, the simulator is run in a sequence of independent trials. To produce deterministic results, Ns-3 use seeds determined by the Application programming interface (API). For multiple runs, this thesis implementation takes advantage of these deterministic seeds by requesting a new seed for each simulation in a series.

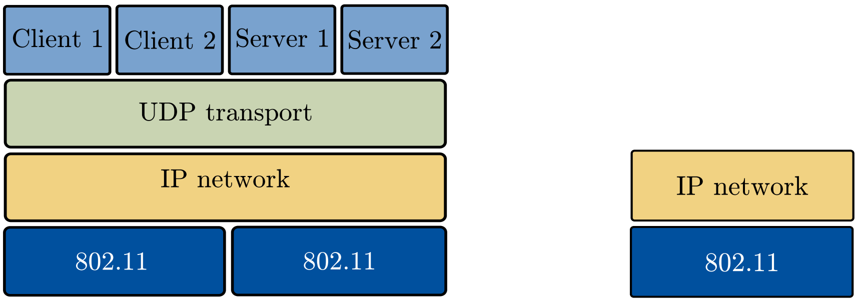
All simulation scenarios are implemented as time terminating systems of nodes in fixed positions running one or more virtual applications. The applications generate traffic that can be transferred over an 802.11b ad hoc network. At the network layer, each node supports IP forwarding through a special routing protocol. This means that policing and classification of traffic can be done in the application, the following routers can thus be reserved from the complexity of traffic classification by enforcing a per-hop behavior. Meanwhile, the routing model is also able to forward according to a configurable maximum number of k ground hops. The maximum k-hop rule, instructs nodes to forward packets destined to a node more than k hops away via the relay, instead of along ground.

3.2 Implementation

The following section will through an overview describe the most important details of both the already implemented parts of Ns-3 and the contributions from this thesis. First comes a summary of the physical entities, before a walk down the most critical layers of the protocol stack serve as an explanation for the logic inside nodes. A complete review of the Ns-3 modules are outside the scope this thesis. Instead, this section provides a minimum of information about the ready-made modules used and a more detailed look at the parts developed solely for this thesis. The source code and utilities used during the research is published together with the report and should be used as a complete reference. Moreover, Ns-3 comes with an excellent online documentation available at [27].

3.2.1 Simulation Entities

Simulations consist of two different kinds of entities described in Figure 3.1. The blocks represent layers from the Internet stack with columns representing models present at that layer. An observant reader might be able to tell that terrestrial nodes are equipped with four application models, a User Datagram Protocol (UDP) stack for transportation, Internet Protocol (IP) for forwarding and WiFi as the link protocol. Relay nodes, however, are not the sources nor the destinations for any traffic in the scenarios and thus they only carry models for the link and network layer.



(a) Layers of the Ground node Model **(b)** Layers in a Relay-node Model

Figure 3.1: The layers of the simulation model

All physical entities are built with the included node model in Ns-3. This model realizes a basic computing device. A node can move in a three-dimensional space according to a mobility model, hold several different net appliances (interfaces) and run application models. The mobility model keeps track of all entity positions throughout a simulation. This model is the basis for determining how good and if a connection is possible between nodes. It is also the basis for interference calculation between transmissions.

With each node holding a mobility object; a node's position and velocity can change based on random variables to simulate scenarios with moving nodes. Since this thesis does evaluate mobility, all nodes are stationary. In order to easily separate nodes that are placed on the ground and the flying relays. All scenarios developed by the author distribute ground nodes in a two-dimensional grid. The only entity that ever have a height above this grid is the relay nodes.

3.2.2 The Physical Model

Ns-3 nodes are equipped with a set of network devices models. In the following simulations, all terrestrial nodes are equipped with two WiFi network devices in different net. The relay entity has only one. Each wifi appliance holds a channel implementation, MAC implementation, and a model of physical properties. Fortunately, since the framework offers a complete 802.11 implementation, it is not too much work to set up a solid WiFi model based on the Yet Another Network Simulator (YANS) described in [28]. YANS offers support for most of the functionality found in 802.11. In the following implementation, the YANS implementation is limited to the 802.11b standard only.

The physical model in YANS is implemented as a state machine with six states present in Table 3.1. Physical layers drop all received bits while in the TX or SYNC

Table 3.1: The states of the physical WiFi layer

State	Description
IDLE	Ready to receive or send
CCA_BUSY	Medium is busy
TX	Transmitting a packet
RX	Receiving a packet
SWITCHING	Changing channel
SLEEP	Interface is sleeping

states. Whereas, for information arriving when in IDLE or CCA_BUSY states, the physical layer calculate the received energy level of the first bit. If the energy level is above a set threshold, it triggers a state change into SYNC and schedules the reception of the last bit. Bits arriving with an energy level of less that the SYNC threshold, but over a set level trigger a transition to CCA_BUSY and the packet dropped.

Channel Implementation

Physical properties of the radio transmission in YANS is described in a channel model. This model implements propagation loss and delay of packets.

As covered earlier, it is unrealistic that packets arrive at all nodes instantly after transmission. Thus, YANS schedules reception of packets according to a model of propagation delay. The model of choice for later simulations assume a constant speed propagation delay with the speed set to the velocity of light in vacuum. Hence, the propagation delay is estimated simply by dividing the distance by the speed of light.

To describe the effect of attenuation of radio signals YANS uses a propagation loss model. With Ns-3 comes multiple models for energy propagation based on the mobility model of each node. These models take the coordinates of a transceiver receiver pair and the transmitters radiated power, and calculate the power perceived at the receiver. The simplest implementation is a scheme similar to the protocol model discussed in Section 2.4. Here, a configured transmission range can be enforced simply by checking the direct physical distance between the coordinates of the originating node and the receiver. Information originating from more than the configured range away is given a high energy loss.

One of the main assumptions of this thesis is that the attenuation of radio signals are higher for nodes that are placed on the ground than those placed on an airborne platform. The argument is that nodes on the ground experience signal loss from obstructions such as trees and buildings. To replicate such behavior this thesis needs a propagation model that takes the height of nodes into account. At the same time, the author expect the attenuation to vary a lot for different kind of environments. Nodes placed in an urban environment might lead to a very short horizontal range and nearly free space path loss straight up. Urban environments might also have a high number of competing systems that cause interference. For hosts positioned in a forest, the situation can change with varying density of vegetation. Including mobile nodes with changing antenna polarity, and the system of changing propagation loss create an interesting research topic on its own.

For the thesis to stay focused on the overall implications of using relay nodes in a MANET it simply assumes that the range to and from flying nodes is indefinite. Based on the constant range propagation model this thesis developed a minimal propagation model that take flying nodes into account.

Table 3.2: Attributes in the propagation model developed for the thesis

Attribute	Description	Default
GroundRange	Maximal transmission range between ground hosts	150 m
AirRange	Transmission range to or from airborne hosts	100 000 m
AirHeight	Minimal height of airborne stations	1 m
PropagationLoss	Loss within range	−95 dBm
TransistionLoss	Loss outside range, but inside transition area	−3 dBm
InterferingDelta	Radius of transition area relative to the range	1

The propagation loss model developed specifically for this thesis take six configuration attributes found in Table 3.2. At the reception of a packet, the receiving interface determines the signal level with a simple distance check, $d \leq r$. If the distance d between two stations are less than or equal to the range r specified respectfully by GroundRange or AirRange, the radiated power minus the PropagationLoss is returned. Stations that fit $d \leq (1 + \Delta) \times r$ with an InterferingDelta Δ, return radiated power minus PropagationLoss and TransitionLoss. Nodes that does not meet these requirements assume a signal level of -1000 dBm, which is way below the sensitivity of the interfaces. Thus, the packet is ignored.

This minimal propagation model provides a simple and consistent representation of signal path loss under the assumption that the range is different for elevated nodes than for nodes on the ground. Note that here there is no difference between two nodes placed at different positions as long as both are within range.

A strict, binary logic is applied; either reception is possible, or it does not exist. Effectively, two stations transmitting at the same time within range will always interfere and force a retransmission.

Based on the power of a received signal at the receiver YANS determines if a transmission was successful. At the scheduled transition from the SYNC state, the interface calls a calculation of the correct reception probability. The probability is determined based on the general noise level, and other transitions that happened between the first and last bit of a packet. To duplicate bit errors that might have happened during the reception the simulator draws a random number between zero and one from a uniform distribution. If the random number is higher than the probability of reception, the packet is treated as if it is unreadable and tossed away.

The attributes of YANS physical model is first initialized with its default configuration before it is set up according to Table 3.3.

Table 3.3: Configuration changes made to the physical model of YANS

Attribute	Default	Set value
RxGain	1.0	0.0
TxGain	1.0	0.0
TxPowerStart	16.0206	0.0
TxPowerEnd	16.0206	0.0
RxNoiseFigure	7.0	0.0

3.2.3 MAC Layer

Unlike in simulation of wired networks, where much of the MAC layer functionality can be abstracted out, wireless networks are relatively unreliable. With multiple stations sharing the medium; collisions lead to lost packets and complex behavior that affect the performance. The simulation must reflect this behavior to stay relevant, and since Ns-3 already comes with a good model, coming simulations use the ready-made model with a few exceptions.

Packets arriving at the MAC layer from higher layers are first queued. Although YANS MAC implementation support QoS, none of the nodes in later simulations are set up with it. All stations are configured with a limited data rate set of only 1 Mb/s over direct-sequence spread spectrum. As a result, the queue is as strict first-come-first-serve implementation, and transfer of data follow a consistent data rate.

Here, the implementation differs from most real implementations of 802.11b. The goal of the thesis, however, is not to simulate a pure and modern WiFi implementation. With this in mind, the current implementation uses 802.11b as an example of a link layer protocol with many of the functions expected from a modern MANETs. Later results have to be seen in this context. A real deployment with higher data rate and support for short PLCP will perform better. Even so, the author expects the fundamental discoveries to be transferable also to a higher rate. A network with rate adoption can only experience higher relative loss in efficiency.

The transmission queue follows the 802.11 timeout procedure described by the DCF [21]. When adding a frame to the queue, they are tagged with the current simulation time. Upon dequeuing, the queue checks the packet time stamp to verify whether or not the packet is too old for transmission. Frames retransmitted multiple times over a period and data that is more than a configured threshold old should be dropped to prevent buffer bloat. All interfaces used in the simulation uses the default Ns-3 upper limit of 5 seconds queue time. To simulate a limited buffer size, the implementation also enforces a maximum number of packets in the queue. Packets arriving at a queue that is more than 400 packets long are dropped.

RTS/CTS is in following simulations configured in the same way as is common for most real 802.11 interfaces. When the interface decides whether a two-way or a four-way handshake should be used, the frame size is compared to a RtsCtsThreshold attribute. If the size of the complete MPDU is larger than the set threshold a four-way handshake is used. The default value of RtsCtsThreshold is set to 2347 bytes. This is more than the maximal size om an MPDU. Thus, only the two-way handshake is used.

Implementation of fragmentation is supported with a simple fragmentation threshold. Any packets bigger than the limit is fragmented into fragments smaller than the fragmentation threshold.

Like most simulations on information and communications systems, the author expects that the network has a transient state prior to the steady state that characterizes a running system. One of the important factors in the link layer is the resolution of addresses between the network layer and the link layer. Address Resolution Protocol (ARP) is a common protocol used to map IP addresses to a MAC interface. In the beginning of a simulation, none of the entities have a record of each other's mapping. The ARP protocol solves this by broadcasting queries on the MAC layer. Hosts configured with the address in question answers the request, and the mapping can be cached for later use. Networks will quickly transfer to a state where many of the nodes have a list of several address pairs. Consequently, the rate of ARP requests will drop over time.

To be able to perform measurements on a system in a steady state without waiting for the ARP cache to build up, a trick is performed by the simulation. Avoiding ARP also reduces the factors that can affect the results of the simulations. Instead of starting each simulation with an empty cache, each simulator scenario begins with an initialization of a complete mapping between all address pairs in the simulation. Through a method developed by the author, the simulator finds all MAC-IP pairs by walking the entire collection of nodes present and store all mappings to a static ARP cache. Next, the cache is installed on all nodes, effectively creating a steady state where all nodes already exchanged address mappings. The result is that no overhead or performance loss is expected to come from the ARP protocol.

3.2.4 Network and Router Modeling

Layer three takes care of information flowing over several hops and create the interface between the low-level node to node communication and the application socket. Ns-3 comes with a fabulous implementation of IP networking. It is very rare to deploy WiFi networking without IP on the network layer. Therefore it is only logical for the following experiments also to use this combination. More precisely, all scenarios are limited to IPv4. The reason for choosing IPv4 over IPv6 is based on the authors experience and the fact that version 6 does not offer new relevant functionality. Besides, the choice of network protocol is not the focus. IPv4 on top of WiFi stands out as a well tested and common combination that should provide relevant insight.

The provided IP model in Ns-3 mimics the Linux network architecture. Each node's network device has conceptually one MAC addresses for each interface. Interfaces may have more than one IP address, and store this information in a list of interface address objects. Just like in Linux, the API has a global configuration for IP forwarding. For the nodes in a network to function as routers and be able to forward information on behalf of others, this is turned on in all scenarios where routing is present.

In compliance with a real-world implementation, the network model needs a routing table for forwarding. It is possible to maintain a static configuration but for mobile networks this approach is impractical. Scenarios, where nodes move in unpredictable patterns, need a dynamic routing protocol to stay fully connected. The dynamic routing protocols provided in Ns-3 concentrates on mimicking existing protocols for research related to the routing protocol performance. Since the simulations described later focuses on general phenomena occurring in wireless networks of mobile nodes helped out by a relay node, it is preferable to abstract out the overhead contributed by existing routing protocols. Besides, no existing protocols known to the author are currently taking flying nodes into consideration.

The following design takes advantage of running inside a simulator with complete control over the environment. While existing protocols in Ns-3, such as AODV and

OLSR uses control messages to discover link state and possible routes, nodes in the simulator have the ability to build a complete picture of placement and expected link state of other devices through the simulator. Link state is found by a configurable expectation of communication range with nodes on the same level and nodes flying above. A configurable time interval triggers node position updates. Based on position and link state, the protocol builds two routing tables with a shortest path first method; The first table considers only nodes placed on the ground and the next, create a complete table including all nodes in the simulation.

The routing tables are implemented as a list of route entries with the destination address used as an index. A route table entry holds the destination address, next hop address, the interface that should be used, the number of hops between the host running the routing table and the final destination. Further, a routing table has straightforward methods for lookup and management of the routes.

Table 3.4: Configurable attributes in the new routing protocol

Attribute	Description	Default
UpdateInterval	Time between new route calculation	1 s
GroundRange	Expected transmission range between ground nodes	150 m
AirRange	Expected transmission range to or from flying nodes	100 000 m
AirHeight	Minimum height of airborne nodes	1 m
DscpValue	DSCP header indicating airborne links	10/AF11
MaxGroundHops	Maximum hops before qualifying for an air link	9999

All available attributes of the routing protocol are described in Table 3.4. At startup, the protocol schedules an update at the configured UpdateInterval. When the schedule calls an update, the protocol purges the current routing table and iterates through all nodes registered in the simulator. For each interface in every node except the one performing the update, the protocol compares the remote network address with all the local interfaces. Nodes with matching network addresses are registered in a list of ground nodes or relaying nodes based on the nodes Z-coordinate for further processing. Nodes with an elevated position above or equal to the AirHeight are here classified as flying. Nodes cannot appear more than once or in more than one list.

The lists form the basis for the routing protocol. It is not necessary for airborne nodes to create a ground table. Thus, they only create one complete table. Dijkstra's algorithm[29] converts the nodes positions and the configured range assumption to a graph of connected nodes. The algorithm finds the shortest path between nodes in a graph and is frequently used for solving problems in navigation or routing. In this approach, it is used on the list of nodes to create a list of the next hop and the number of hops needed to all other nodes. Finally, this list is converted into IP-routes and placed in the respective routing table.

The first step in the implementation is to initialize three lists. In the pseudocode found in Listing 3.1 `Q[v]`, denotes the queue of nodes `v`. `dist[v]`, is the logical distance (hops) to a node, and `prev[v]`, represents the previous node (neighbor of `v`) through the current shortest path. When the algorithm is called, it iterates over all nodes ❶. `Q` is filled with all nodes. If the node `v` is not the node performing an update, the logical distance is set to infinity. `prev[v]`, is set to be the node itself.

The second step is to pick the node with the shortest distance ❷. For the first run, this will always be the one running the route update (`myself`). This node is now stored in a temporary variable, `u`, and removed from the queue. Now the algorithm finds the physical distance between the selected node and all other nodes. `txRange()`, implement the expected range ❸. If the physical distance from `u` to `v` is less than the GroundRange or one of the nodes in questions is airborne and the distance is less than AirRange; it checks if the logical distance to the selected node plus one is less than the distance stored in the distance list. If it is, the distance and previous list is updated accordingly.

The `dest` and `prev` are updated until the queue is empty. Note that another approach could be to add the physical distance or a link metric instead of one ❹. The result would be that physically close nodes or nodes with a low sum of metrics is preferred. The physical distance approach allows nodes to forward information over paths with lower free space propagation loss. While, if the metric represented packet loss or an indicator of mobility, the most robust path would be chosen. The metric cold even be a representation of capacity to choose the path with highest available capacity and maximize load distribution. On the other hand, since the propagation model do not emulate path loss inside range and the nodes are stationary, such approach is futile. Besides, choosing links based on fluctuating metrics will demand that a real network exchange control traffic more regularly.

Anyhow, when `Q` is empty, this marks the end of Dijkstra's algorithm. The final step is to walk from all nodes except `myself` with a distance of less than infinity, find the first next hop (closest neighbor of `myself`) and create a route entry ❺. Up to this point, the protocol has only been concerned with node entities. To convert entities to route entries the algorithm needs to obtain the IP address of the destination, the next hop and the correct interface for communicating with the next hop ❻.

A callback registered by the routing protocol are notified with local interface changes. This way the routing protocol can iterate over all local interfaces. For all registered local interfaces, the protocol iterates over interfaces present at the next hop node. If the IP network address of two interfaces matches, a route entry are created to

Listing 3.1: Pseudocode of Dijkstras algorithm

```
  def dijkstra(nodes):
❶     for v in nodes:
          Q.append(v)
          if v is myself:
              dist[v] = 0
          else:
              dist[v] = infinity
          prev[v] = v

      while Q is not empty:
          tmp = infinity
❷         for v in nodes:
              if Q[v] and dist[Q[v]] < tmp:
                  u = Q[v]
                  minDist = dist[v]
          del Q[v]

          for v in nodes:
              if v is not myself:
                  d = findDistance(v, u)
❸                if d < txRange(v, u):
❹                     tmp = dist[w] + 1
                      if tmp < dist[v]:
                          dist[v] = tmp
                          prev[v] = u

      for v in nodes:
          if dist[v] > 0 and dist[v] < infinity:
              tmp = v
❺             while dist[tmp] > 1:
                  tmp = prev[tmp]

              prev[v] = tmp
❻             addRoutes(v, prev[v], dist[v])
```

every interface present at the destination node via the matching address and local interface.

To avoid nodes with a separate interface intended for relay communication add this interface as a route for ground communication a check for longest bit match is performed on the interface addresses. The match adds two IP-addresses and count the number of ones in the product. If a route exists in the routing table with a higher number of matching bits the new route is ignored.

It is important to note that since the algorithm iterates over the node interfaces when creating the routes for the table; the network interface index matters. The order an interface is initialized in a node determine the interface index. Thus, care should always be taken to initialize the interface intended for ground communication before interfaces destined for relay communication. Under the assumption, that relay nodes have a longer range than ground nodes and that the relay is not capable of directing energy, ground paths should be prioritized when sharing the same channel. In most cases it is beneficial for the network that a ground node with a short range interferes than if the flying relay interferes with a bigger part of the network.

Route lookup

Whenever transport protocols hands segments down to the IP layer for unicast communication, the routing protocol is asked to output a route as if querying the route cache. Here the routing protocol can enforce an interface for output of the segment. Caching simply a lookup in the correct routing table based on the Differentiated Services Code Point (DSCP) header field set by the application layer. The returned route indicate the right output device. Destinations not found in either of the routing tables triggers an error in the transport layer socket telling the application that the destination is unreachable.

For the routing protocol to handle packets intended for the flying relay correctly, the packets need to be marked in some way available to the routing protocol at the network layer. Traditionally, QoS implementations in IP-networks use two approaches. One way to mark packets is to append a tag to the packet like in Multiprotocol Label Switching (MPLS). Another method involves the Type of Service (ToS)-field already in the IP-header such as in Differentiated Services (DiffServ). The benefit of using the ToS-bits is that new information does not need to be added to each packet, with less overhead as a result. In the application described later, traffic classes are marked through DSCP set in the ToS-field. The routing protocol keeps a configured tag found in the DscpValue attribute used to symbolize packets intended for a relay path.

Route lookup first checks the traffic class given by the IP header. If the DSCP does not equal to the configured airborne relay tag, the routing protocol first looks for a matching route in the limited ground routing table. Whenever the lookup for a route does not return a valid route, the relay routing table is checked. However, if an incoming header suggests the relay routing table, and it does not return a valid route, it is no reason to check the ground routing table. The relay routing table contains all nodes both flying and terrestrial; thus, the ground-only table cannot contribute.

Datagrams intended for a ground path can, however, use the table with airborne relay nodes if the best path suggests a path with more than a configurable k hop limit, MaxGroundHops. By setting an upper bound on router table hops, it is possible to experiment with scenarios where airborne relays offload the ground network by transferring a packet directly instead of being forwarded through a long path along the ground.

3.2.5 Application Modeling

In order to simulate a source of information and a provider of realistic network load, an application model is implemented. With a simulated application in Ns-3, it is trivial to install it in the different nodes to create different communication scenarios. The developed application is an extension of the on-off application already in Ns-3. In this application is a constant bit rate application which can be turned on or off. The on-off application model a finite-state machine with two active states. After an initialization the application shifts between an on and an off state. State transitions occur in compliance with a schedule of two random variables OnTime and OffTime. All available attributes are described in Table 3.5

In the initialization phase the application creates a socket for communication to all entities. The approach taken is to loop over all hosts and create a socket for each. By trying to bind to the first interface present in all nodes, the application obtains a list of socket candidates. The sockets are all set up with the configured DSCP attribute. Sockets not able to bind properly are logged, but no attempt of reconnection or keeping the status of a socket is performed. Thus, care should be taken to always start a server application at all possible values of the random variable RemoteNodes prior to client initialization.

First, the application enters the off state, until the first off-timer runs out. In the on-state, a random socket from the list of socket candidates is picked. A empty packet with a size of PacketSize is then sent to the socket for transport. To simulate a constant bit rate, a next packet is scheduled DataRate^{-1} seconds after the first. Since the duration of both states depends on random variables, synchronized flows can be avoided.

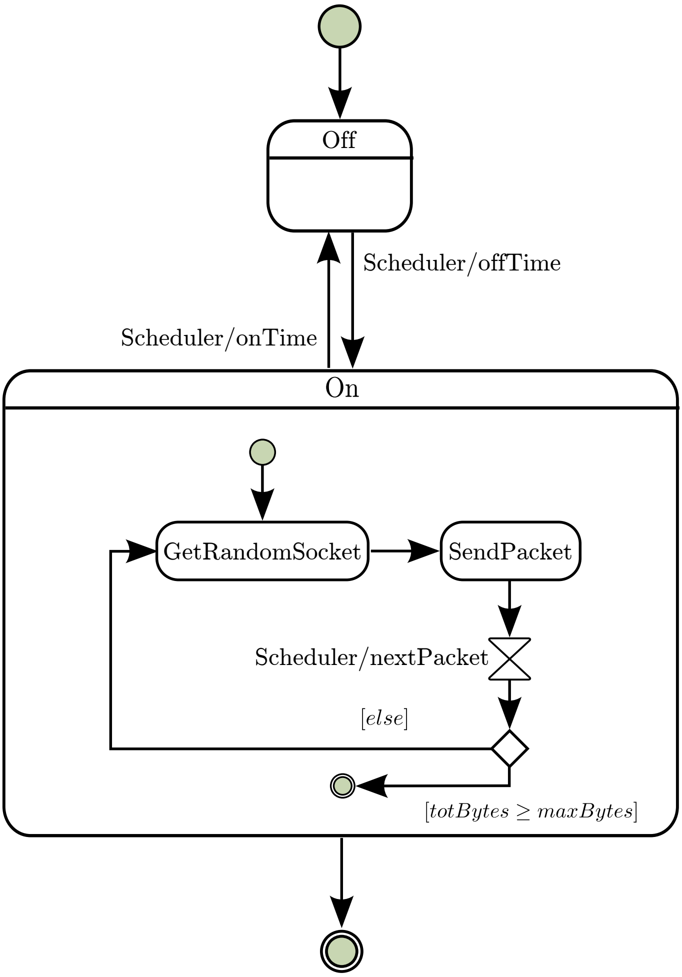


Figure 3.2: Application state diagram

Table 3.5: Configurable attributes in the application model

Attribute	Description	Default
DataRate	Data rate when in a on state	1000 kb/s
PacketSize	Size of packets in bytes	512 bytes
RemotePort	Port open at server	9
RemoteNode	Node index of remote host	1
OnTime	Duration in on state	1 s
OffTime	Duration in off state	0 s
MaxBytes	Maximal number of bytes to send	0 bytes
Protocol	Socket type to used	UDP
DSCP	Type of service identifier	0/Best effort

When the OffTime run out, a new empty packet of configured size is created, a random socket is drawn from the set of sockets and the chosen peer set in the destination field before the packet is handed over to the transport protocol for further transportation. After each packet, the application checks the sent number of bytes, if it is more than the MaxBytes limit, the creation of packets stop for the rest of the simulation like described by Figure 3.2. A MaxBytes equal to zero results in infinite bytes to be sent until the simulator is stopped.

With all ground nodes running two instances of the application model, set up with different attributes, and a random or constant value, the network can be tested under multiple distributions of traffic load.

3.2.6 Statistical Variables

An important part of the simulation is the gathering of statistics and measurements on the system model. For later discussions, the term observation represents one of all the possible outcomes of a measurement, and the term sample is used to describe a set of observations. When the observations are independent and identically distributed, the sample is a random sample.

Network systems are typically non-terminating systems. When simulating such systems, the initial conditions and termination conditions must be specified in each experiment. For steady state analysis, it is important that the experiment last long enough to represent a steady state of the system and data must be collected after the system has reached a steady state. For instance, when simulating highly loaded interface buffers, the initial fill level is typically lower than the average fill level. Hence, using the measurements from the first part of the experiment might change

Table 3.6: Statistical data gathered in a simulation run

Measure	unit
ground nodes	-
data rate of application	kb/s
packets sent	-
packets received	-
bytes received	-
average hops	-
average transfer time	µs
average physical distance	µs

the result and not reflect the steady state behavior. To avoid biased results, the observations must start after the transient period. [30]

The developed simulation models have taken the above considerations into account. Where possible, transient factors are abstracted out or performed outside of the schedule so that they do not affect the measurements. Another factor that is important is the routing table as well as ARP cache. Since simulations performed in this thesis do not include mobile nodes the routing table does not change during simulation, thus the easiest solution is to schedule a routing table update prior to sample collection. As mentioned before, the ARP cache is filled prior to simulation start. Buffer load, or queuing, can however not be abstracted out. This is considered an important property of a MANET and with changing traffic patterns it is interesting to include this property into the simulation. To avoid biased results from the period where the queue is filling the first simulation period should be avoided.

For each run of the simulation, the applications store statistics from the simulation in a file for later processing. Table 3.6 gives an overview of stored values. All nodes do also run an uncomplicated server application for each client application. The server registers packet arrival, calculate statistical data and stores it for succeeding analysis, before dropping the packet. In addition, the simulator supports a complete packet capture at each interface. Data provided by the application can be used to calculate the number of routers visited, the delay and the physical distance traveled. The number of routers visited can be calculated by comparing the TTL field in the IP-header of packets. Determining the end-to-end delay is done by taking the time of arrival minus the time of creation. The distance between the sending node and the receiver found by looking up the position of the sending interface and the receiving interface give information of the direct distance traveled by the packet under the assumption that the original node has not moved during transfer.

Instead of plotting results from the simulation directly, a computer script calculates an interval estimator for the observed samples. Assume that a simulation returns n independent and identically distributed values $\{x_1, x_2, \dots x_n\}$. When observing a random variable, x will vary between measurements. A set of observations can be represented as a mean of all measurements, $\bar{x}$. This is called a point estimate. Instead, an interval estimator specifies a range in which the measurement is estimated to lie. A confidence interval is a common way to report observations. Together with a point estimator the interval creates an upper and lower confidence limit. The probability that the confidence interval includes the unknown mean value is specified by $(1 - \alpha)$ and is called the level of confidence. This provides a degree of reliability of the estimated sample. The following measurements use a 95 % confidence interval where suited.

Since the sample distribution and thus the standard deviation of the samples are unknown, Student's t-distribution is used as the critical value t. Here, t is the $\alpha/2$-quantile with a $n - 1$ degrees of freedom found in Student's t-percentile table. The estimated variance of $\bar{x}$, S^2 and an estimated confidence interval is then found by:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.1}$$

$$\left(\bar{x} - t\frac{S}{\sqrt{n}},\ \bar{x} + t\frac{S}{\sqrt{n}}\right) \tag{3.2}$$

A more elaborate explanation is provided by Iversen in [16, Ch. 13].

Chapter 4
Experimental Studies

This chapter presents simulation scenarios and results found during the simulation. To be able to illustrate interactions and effects present in MANETs the chapter starts out with a simple scenario and add complexity to create several interesting situations more likely to be relevant in real situations. Inspired by [3], the first scenario places nodes within range to find a baseline for performance. The next scenario uses a chain of nodes to determine how longer paths affect the throughput. A chain of nodes does also create the basis of an example where a ground network can outperform a relay, even with more hops.

To take the final step from a strict placement in only one dimension; the chapter finishes with nodes spread evenly in a grid. The relay is still present and hovers in the center of the square universe. Rather than one or a few fixed traffic sources, in this network, all nodes contribute by sending traffic to a randomly chosen destination.

4.1 Scenario Descriptions

For reproducibility, the thesis is published together with all scenario files used. Together with the scenario files follows also a series of terminal scripts used to automate multiple runs and evaluate the difference between various simulations. To reduce simulation time the scripts support multiple runs running in parallel to take advantage of multi-core processors. Where relevant, the scripts calculate a mean of multiple runs and calculate a 95 % confidence interval. In cases where nothing else is described, scenarios use the default values presented in Chapter 3.

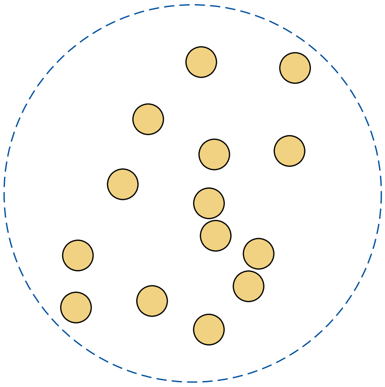


Figure 4.1: Single cell of nodes

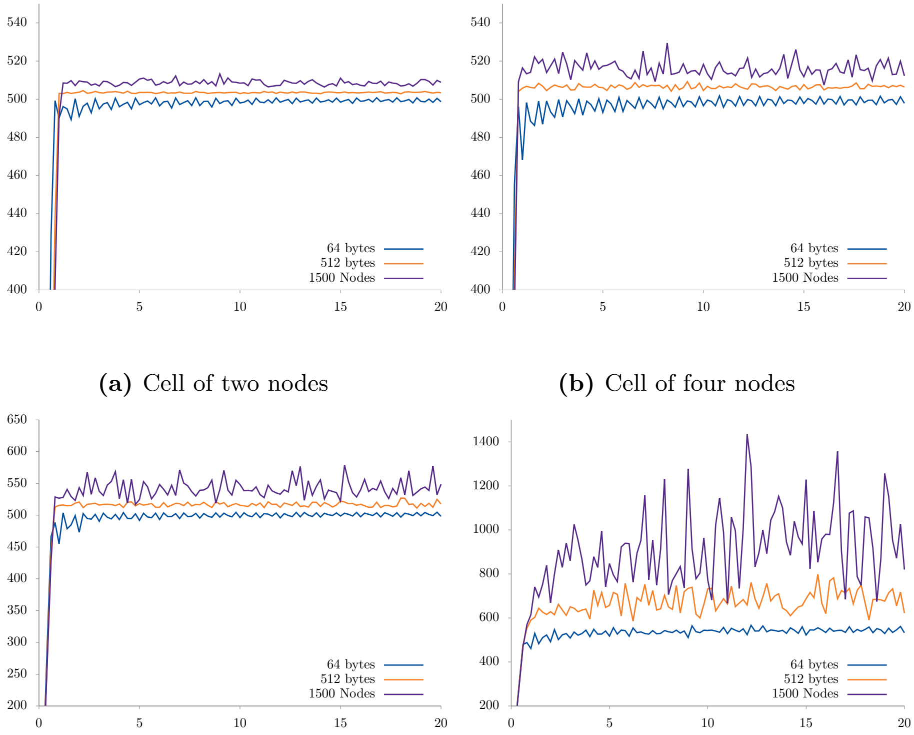
4.2 Capacity When All Nodes Are Within Range

As a baseline for comparison of following situations and as a validation of the simulator; the first case consists of a circular area with a diameter equal to the transmission range. Nodes are spread uniformly within this circle, creating a single broadcast domain illustrated in Figure 4.1. Each node is a source of traffic with a greedy sender strategy. To achieve a traffic pattern like this; all nodes run one instance of the thesis' application. DataRate is set to a constant rate of 1 Mb/s, and RemoteNode is a uniformly distributed variable with values within $[0, N]$. The radius of the circular area is 150 m.

It is important that the interface queue has reached a steady state before starting the measurements. To find out when the system is in a steady state the first simulation aims to see when the delay of packets stabilizes. The charts in Figure 4.2 show how the delay of packets changes throughout simulation time. The units on the X-axis is seconds and milliseconds for the Y-axis. Values for the changing delay is found by measuring the average delay of received packets in 200 ms intervals for cells of a different number of cells and packet size. The samples are averages from five different simulation runs.

From the beginning of the simulation, it is clear that the delay increases sharply with a filling of queues. At 600 ms the delay changes to more consistent values. Since the application creates more packets than the WiFi interface can handle, the queue grows. Longer queue leads to longer queuing delay. When the queue length reaches 400, new arriving packets are dropped. With tail-dropping of packets due to a finite queue size, it is natural that the queue delay is proportional to the buffer size.

For two nodes the steady state seems to appear instant when dropping of packets



(a) Cell of two nodes

(b) Cell of four nodes

(c) Cell of eight nodes

(d) Cell of fifty nodes

Figure 4.2: Packet delay of received packets over time in a single cell of nodes spread within range of each other

starts. A packet size of 512 bytes appears as very stable. There is fluctuation in the delay after the first period. However, the distinct rise in delay seems to have ended. For five and eight nodes, another artifact is present for 64-byte packets. Here, a series of dips in delay is visible, but before 5 s has passed all packets seems to arrive in a stable state. It should, therefore, be safe to assume that the transient state is over and a steady state for this scenario is obtained well before 10 s has passed. The later simulations throw away samples up to this point.

Figure 4.3 shows how the total throughput responds to an increasing number of nodes. Over a period of 30 seconds, the simulator measures the throughput received by all nodes. By measuring the sum of bits received by an application server running in a recipient during a simulation a clear image of available capacity for each node inside a single cell is obtained.

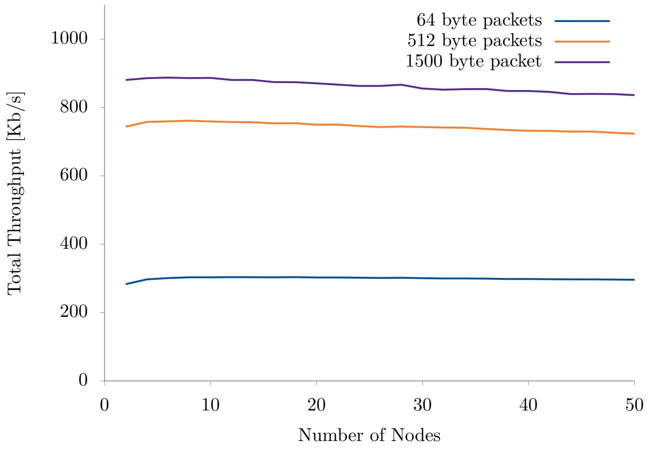


Figure 4.3: The total throughput of nodes spread within range of each other

The results differ from the results published by Li et al. in [3]. In the article, Li simulates and measure a physical network to have a clear loss in throughput with increasing number of nodes in range. However in [2], Duda state that the total throughput of a simulated network like this stays nearly constant. Contradicting to what might be obvious, Duda notes that a slight increase in capacity can be seen with an increasing number of nodes. This is due to that an increase in the number of nodes also results in a shorter interval between transmission attempts. Even though the higher number of nodes leads to a higher number of collisions and that colliding nodes increase their contention window, increasing the number of nodes can also result in an unfair distribution of capacity. Nodes that have not collided lately have a smaller contention window and can take advantage of the colliding node's higher contention window. In sum, the total capacity is decreasing, however not that much. Even with fifty nodes competing for capacity with 1500 byte packet the decrease is only 50.96 kb/s when compared to two nodes.

For a network of N nodes where the total throughput is constant, the per node throughput will follow $O\left(\frac{1}{N}\right)$ — which do not come out as promising when it comes to the scalability of wireless networks. A higher number of nodes result in more time spent in backoff, higher delay and less throughput available for each node. Nevertheless, later scenarios can provide some hope. Concurrent transmissions made possible by path loss will be able to increase the per node capacity.

An important takeaway from the results in Figure 4.3 and the efficiency calculated in the background chapter is the difference in performance of large and small packets. In applications without real-time constraints which regularly transmit small packets, it is beneficial to buffer information sent to the same destination until packets can be filled.

4.3 Capacity In a Chain of Nodes

The first step in the direction of a proper MANET is a simple chain of nodes. In the same way as traffic is forwarded in a real MANET, a chain of nodes allow packets be transmitted between intermediate nodes to reach destinations outside of the originating nodes coverage. The following scenario uses the first node of a chain of nodes as the only source of traffic. Node one creates packets at a fixed rate addressed to the last node in the chain. Ground nodes are distributed by a distance of 150 meters with a range such that all wireless ground stations can transmit to its nearest neighbor, but not further. However, with an InterferingDelta of one, transmissions are received just enough to interfere out to the second neighbor. The loss is enough to interfere with ongoing communications but less than the sense threshold that triggers the interface to enter a busy state. The result is that collisions can occur up to two hops away while reception can only happen over one hop.

Remember how Section 2.4 and Figure 2.6 used geometric arguments to estimate exactly such kind of network. Based on the assumptions found here the expected throughput should be best at a network of only two nodes and a decreasing throughput with increasing number of nodes. Since only one out of four nodes in such networks can transmit simultaneously (See Figure 2.6, it is further expected that the ground nodes in long chains at maximum can transfer data to the last node at one-fourth of the capacity available in each network interfaces.

Figure 4.4 show simulation results for a chain of nodes sending 512 byte packets over a period of 30 s. Simulation is performed with increasing DataRate set in the client application over multiple runs. The graph shows how the received data rate at the end changes linearly with increasing load in by the first node. It is clear that the networks manage the data rate nicely up to a point where the linearity stops. The maximum Throughput varies significantly with more nodes in the network up to five nodes. For networks with more than five nodes, the throughput response stays roughly the same. For eight nodes, it is also possible to see how 802.11 fails to find the optimal rate for a longer chain like noted by Li et al. [3]

The WiFi protocol is not able to find the optimal rate because the originating node does not have insight into the traffic situation outside of range, The source node's transmitting rate then become too high for the rest of the network to handle. The output rate is determined by the experienced traffic.

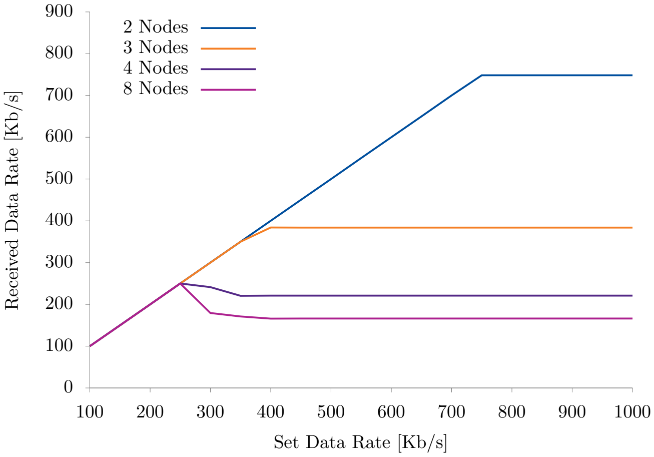


Figure 4.4: Received throughput at the last node in a chain of nodes given a data rate created by the first node in the chain sending 512 byte packets.

The originating node sends packets at a higher rate that what is possible to forward in a central position with more neighbors. Without the RTS/CTS handshake used by Li; the impact is less apparent, but still present.

At a data rate of 250 kb/s the network can keep up even for a chain of eight nodes. For more than four nodes, higher load results in a decrease in capacity. Here the system only performs around 166 kb/s. With chains longer than eight, the throughput seems to stabilize with a maximum throughput of around 160 kb/s. An increasing number of nodes still lead to lower performance, but not as significant as found in chains of less than four nodes. The chain of two nodes peaks at 748 kb/s for 1500 byte packets. The reason that two nodes do not perform the full 1 Mb/s is because of overhead. Back in Section 2.5.1, an upper bound efficiency of 0.77 was found for a WiFi channel transferring 512 byte packets. The upper limit expected throughput should thus lay around 770 Mb/s, which is not too far from the measured 748 kb/s.

The results seems to confirm the theoretical upper bounds described earlier. For chains of nodes, the upper bound throughput follow $O\left(\frac{1}{N-1}\right)$ for $N \leq 5$ and $O\left(\frac{1}{N-1}\right)$ for $N > 5$ when the chain has only one source placed at the end and transmitting to the other end.

Samples in Figure 4.4 have little variation and a very tight confidence interval. For readability, the confidence interval is not included in the plot.

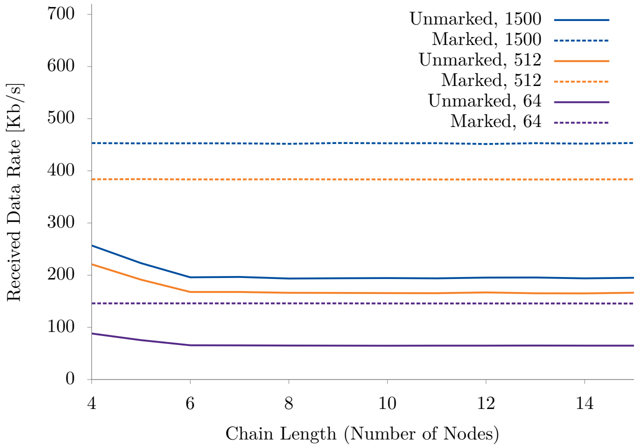
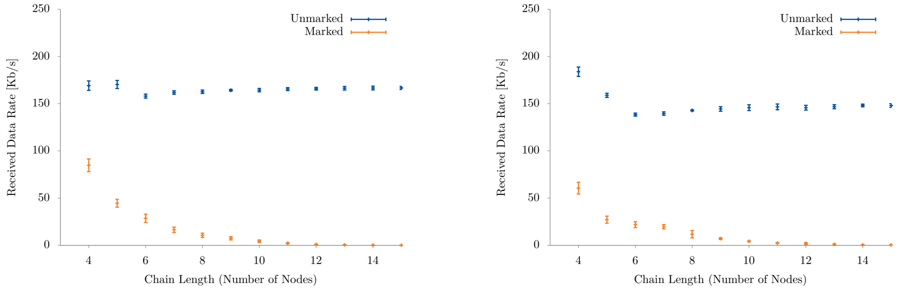


Figure 4.5: Received throughput at the last node in a chain of nodes with traffic distributed between an airborne relay and terrestrial nodes with two separate channels

Comparing throughput Over a Relay and Along the Ground

For a network assisted by a relay node where the ground nodes use a separate, interference free channel for relay communication, one can expect the network performance for the ground network to be unchanged even at longer chains. Traffic over the relay should perform like a chain of three nodes. Since the number of air-hops stay the same. The increasing distance will contribute to delay, but for the distances present here the propagation delay is small. The chart in Figure 4.5 shows the result of a simulation of a chain of increasing number of nodes. Node one sends 1500 byte packets in two streams with a constant rate of 1 Mb/s. The first stream of packets are tagged with a DSCP header indicating a relay handling, and the other is not marked.

The airborne relay hovers over the network in a height of 10 meters and has a range that covers the entire system. For a chain of three or fewer nodes, both traffic classes will be directed to the terrestrial links since this path is considered the shortest by the routing protocol. Here it is expected that the sum of both classes is equal to the maximum throughput found in Figure 4.4. Depending on which traffic class that is initialized first one class can outperform the other by being first to put packets into the output queue of the terrestrial interface. The high rate created by the applications quickly fills the queue and effectively blocking the interface. Thus, the results vary. Over an average of many simulations, however, the result is like shown in the chart. For longer chains this is not an issue. Here the two applications are directed to their dedicated interface by the routing protocol. Thus, they fill one queue each and do not intervene. Notice how the throughput of marked 512 byte



(a) 512 bytes packets without RTS/CTS **(b)** 512 byte packets with RTS/CTS

Figure 4.6: Received throughput at the last node in a chain of nodes with traffic distributed between an airborne relay and terrestrial nodes with both sharing one channel

packets matches the value found in Figure 4.4. Traffic forwarded over the relay node matches the throughput for chains of three nodes

A different scenario evolves if limiting the interfaces to use one shared channel. The chart in Figure 4.6 shows what happens with the throughput. The horizontal bars denote a 95 % confidence interval on the throughput measurements obtained over 30 seconds. Chains of more than three nodes are the only scenario where the relay is used. Here the results have changed. Where the relay outperformed the terrestrial network when given a separate channel. It is clear that under a scenario where ground nodes and air nodes compete for access to the medium; ground network outperforms the airborne. What seems to happen is that ground nodes with a shorter range can transmit at the same time outside of range. This spatial reuse makes it nearly impossible for the relay node to enter the idle state. The relay that has less power attenuation can receive transmissions from the whole chain of nodes. For each received transmission the relay interface is placed into a CCA_BUSY state. Even though node one senses that the medium is ready and transmits to the relay, the relay is in a busy state caused by transmissions happening at the other end of the chain. In turn, the missing ACK from the relay forces node one to increase the contention window. Effectively, creating an even lower probability that packets arrive at the relay in an IDLE state.

While this is related to the hidden node problem, a simple RTS/CTS handshake like implemented in 802.11 is not going to solve the problem. Figure 4.6b demonstrates this clearly. Node one is still not able to receive RTS from a node more than one hop away. If the relay, however, retransmitted all CTS and acknowledge packets it received from ground stations the nodes would know that the relay is busy. Yet, it is likely that the RTS reception in the relay would collide with a concurrent transmission

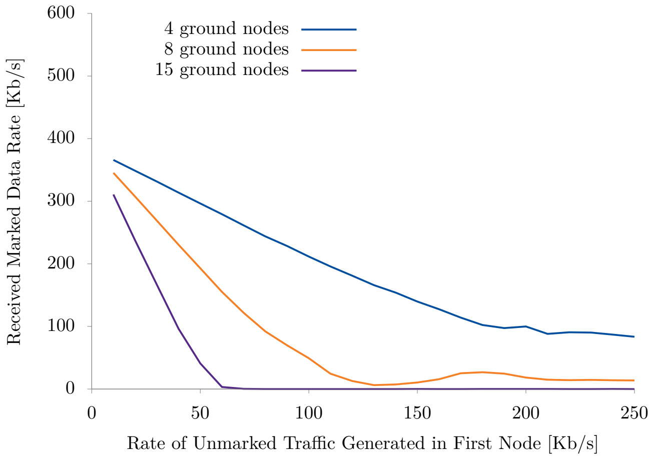


Figure 4.7: Received throughput at the last node in a chain of ground nodes with traffic distributed between an airborne relay and terrestrial nodes, both with 512 byte sized packets

from another ground node. The problem of using one channel is revisited in Chapter 5. Here, a purposed change of the MAC-protocol allow relay nodes to regularly silence the entire ground network and invite terrestrial nodes to transfer via the relay.

Different Ratios of Load Between Marked and Unmarked Traffic

A network with a constant equal load of marked and unmarked traffic might not be a realistic scenario. A typical deployment would perhaps run a small portion of applications that need low delay which is suited for forwarding over the airborne relay in a busy ground network. As seen in Figure 4.7 such a scenario is even less suited for a network with one common channel. The chart show throughput in a chain of nodes where a rate of 1 Mb/s is created in node one and marked for relay transfer. At the same time, unmarked packets are generated at varying rate. The X-axis show the rate of unmarked packets and the Y-axis show the received throughput of marked packets in the last node in a chain of four, eight or fifteen nodes. By varying the load in the ground network, it becomes apparent that the relay responds poorly to increasing load in the ground network. It is also evident from the charts that an increasing number of ground nodes reduces the relay capacity.

The realization that long range leads to more nodes contending for the wireless medium is also made by Landmark et al. in [31]. Through simulation, the paper shows how a MANET of nodes with multiple radios running a proactive routing protocol behaves in scenarios with different loads, and where only a subset of the nodes are equipped with long range radios. Such scenarios can typically be that nodes

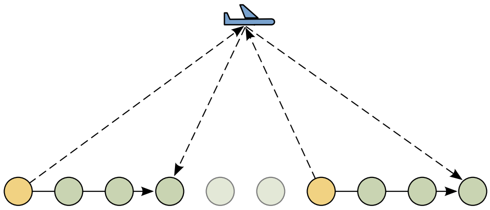


Figure 4.8: Different paths in two chains of nodes

are a combination of vehicles and humans walking by foot, where the extra weight of carrying two wireless interfaces might be best suited for vehicles. Motorized vehicles can also be equipped with antennas that better enable relay communication. A scenario where all nodes cannot connect directly can also rise from varying vegetation or other obstructions.

In contrast to the simulation performed during this thesis, Landmark et al. models mobility. With mobile nodes, it can be an advantage to use long range links to stay connected. However, a varying long range connectivity can lead to inconsistent routes in the network. Results from [31] show that packets might be lost as a result of inconsistent routes, forwarding packets until the TTL is exceeded.

4.3.1 Lack of Spatial Reuse in The relay path

From a logical standpoint and under the assumptions made in Section 2.6, the predicted throughput of a relay shared by two traffic sources should give a maximum utilization of $\frac{1}{4}$ where two chain sets perform $\frac{2}{3}$ for four nodes, or $\frac{2}{4}$ for chains longer than four. The situation becomes be even worse when thinking about collisions between nodes using the relay. The seventh node might perceive the channel to be free because it is outside of the range of the first node and send information to the relay at the same time as the first node — creating a perfect example of the hidden node problem illustrated in Figure 4.8.

If the first node sends information to the fourth, and the seventh node transmits to the last node; both sources carry over three hops without interference with each other. The two nodes not participating between the fourth and the seventh node will, however, experience the load of both, but assume that they do not contribute to traffic or do not exist. Even though traffic marked for relay transfer only travel over one hop compared to the three hop of terrestrial packets, Table 4.1 show that the total capacity of the ground network is higher.

As presented in Chapter 2, the hidden node problem can be (partially) solved with the implementation of RTS/CTS already implemented in 802.11. Results in Table 4.1 support this. At the same time, the four-way handshake introduces more overhead.

Table 4.1: Total throughput in a optimal chain of ten nodes with two traffic sources measured in kb/s at the destination with a 95 % confidence interval in parentheses

	Marked	Unmarked
RtsCtsThresold=0		
64	105.30 (105.74, 143.84)	143.84 (143.38, 144.29)
512	341.03 (341.85, 450.47)	450.47 (448.00, 452.94)
1500	428.24 (429.93, 558.16)	558.16 (553.68, 562.63)
RtsCtsThresold=2347		
64	136.72 (135.99, 137.46)	176.41 (175.81, 177.02)
512	271.55 (268.30, 274.80)	443.21 (440.78, 445.64)
1500	206.12 (201.80, 210.43)	516.76 (512.40, 521.11)

Unmarked traffic in the experiment is affected by the same RtsCtsThresold. Here, the increased overhead can be seen as a reduction in throughput. The table show that the throughout for marked packets of 512 and 1500 bytes improves significantly. Small packets however, performs even worse with the extra overhead.

To reduce the cost it is possible to use the four-way handshake only for large packets like the RtsCtsThreshold is intended. The argument is that large packets have a higher probability of colliding. At the same time, the gain of avoiding collisions is reduced if the handshake takes more resources than retransmitting until completion.

4.4 Random Traffic in a Lattice Network

Previous analysis shows how successive nodes in chains of nodes interfere with each other. A topology of an absolutely straight line with nodes strictly placed at the range of each interface is not very realistic. To further measure the effectiveness of ground networks assisted by a relay this section consider a lattice network. Actual networks have nodes that interfere at more than one edge, with changing degrees of density. Consider a scenario where nodes are evenly spread in a square area. The optimal placement is with nodes spread with just a receivable distance like in the chain situation. A less dense network will not be able to communicate and with nodes nearer to each other the interference increases. Motivated by the discussion on scaling traffic patterns in [3], it seems worthwhile to reduce nonlocal traffic. Not only can a relay offer a short path, it is likely that long paths cross the center of the network. Thus, a relay might reduce the load of the busiest part of the network.

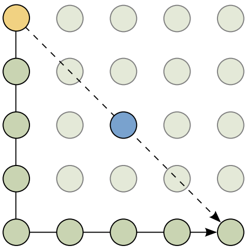


Figure 4.9: Comparing the worst case path length in a 5 × 5-grid between relay and terrestrial path

The following scenario takes advantage of the maxGroundHop attribute found in the routing table to enforce a maximum of k-hops. Packets destined for a ground node with a route entry more than maxGroundHop hops away are allowed to resolve routes in the relay routing table, and thus receives the shortest possible path.

Let all nodes be a source of equally sized packets with a random destination. The expected path length is affected by the number of nodes, and with increasing number of nodes, the delay is expected to rise. Figure 4.9 illustrates how the relay might function as a shortcut through a grid of nodes. The illustration shows a 5 × 5-grid. The worst cases scenario in respect to delay is when two nodes on opposing sides of the network. For a 5 × 5-grid, packets can experience up to 8 hops. In Figure 4.9 the node in the center represents a relay with complete coverage over the network. As illustrated, this gives the opportunity to only use two hops.

From Figure 4.4 it is clear that the highest performance is found between two nodes communicating directly. However for the network to stay connected, it is not acceptable to just drop packets destined far away. Grossglauser and Tse [18] presents a two-hop relay routing protocol where relay nodes are used to store information destined to nodes out of range. For applications with loose delay constraints, they find that mobile relay nodes can increase the average throughput. An airborne node could even approach its destination faster than a ground relay. However under the assumptions held by this thesis, the relay can even transfer directly without movement. Instead, the MaxGroundHop attribute from the developed routing protocol is used to reduce the number of hops and maintain the local traffic pattern between ground nodes.

However, like in the scenario with two chains of nodes, the relay cannot handle the traffic from all nodes in the network. Since the relay operates in one broadcast space, many contending ground nodes lead to long queues and a higher probability of packet loss due to collisions. It is possible to decrease the probability of collisions by allowing the use of terrestrial paths, which take benefit from spatial concurrent transmissions. Thus, the delay in a grid of nodes with a max k-hop routing approach turns out to be an optimization problem.

Reducing the number of hops increases the probability of collisions at the same time as it increases the queuing delay of relay interfaces. While increasing the number of hops results in longer paths and a potential high total delay from experiencing contention and delay over multiple nodes. It is also important to consider that the establishment of a route from a source node to a destination node requires a simultaneous availability of all nodes between the source and destination. Mobility can also cause route failures, especially in sparse networks. Increased range in the relay leads to less frequent route updates and less loss caused by route errors.

The approach used this far to evaluate the maximal performance is of no use in a random lattice scenario. Nodes creating packets at a high rate will quickly fill the interface buffers. With full buffers, new arriving packets will be dropped on arrival. Consider also how the rate of packets arriving from a local application is much higher than from neighboring nodes. The result is that packets will be forward very few hops, and the measured throughput will not come from packets traveling over multiple hops. Simulation on this approach shows that the mean hop length of received packets for a grid of 100 nodes, is less than two. This means that the uniform distribution of destinations is reduced to only nodes two hops away. Most other segments are lost. Thus, to gain insight into maximal throughput of a grid network, there is a need for another approach. One option would be to implement an application aware of the queue length, that reduced the creation rate to allow forwarding of arriving packets.

Another option is to use Transmission Control Protocol (TCP) as transport protocol where rate control for a fair sharing of interface buffers is already supported.[32] Here, a congestion control protocol uses a number of mechanisms to provide high performance while avoiding congestion collapse and provide a fair allocation of network resources. Like in 802.11 TCP uses ACK or lack of ACK to gauge the network condition between two hosts. The additive increase/multiplicative decrease TCP controls sending applications rate, and will give a fair allocation of resources. [33]

Using TCP in MANET does come with its challenges. Mirhosseini present in [34] that the assumption that all packet loss is due to network congestion is not true for ad hoc networks. Packets loss is just as likely to come from routing error or poor connection. A solution would be to implement the ability to distinguish the reason for packet loss, or reduce the probability of non-congestion packet loss. With an improved TCP protocol that only detect congestion if a set of metrics support that packet loss comes from congestion, Mirhosseini shows a significant reduction of the probability of false detects. However, choosing a solution to improve the

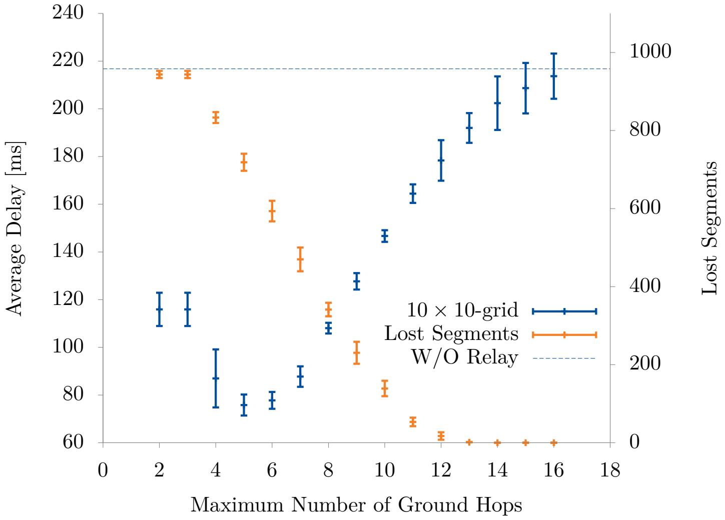


Figure 4.10: Delay of a slow rate of traffic sent with random destination in a grid of nodes without RTS/CTS

TCP performance is complex and getting the changes adopted is challenging. Larsen suggests in [35] for this reason to use UDP and handling congestion and reliability in the application layer.

Simulations using TCP for transport or implementation of congestion avoiding applications is outside the time frame of this thesis and is left for later studies. Instead, simulations uses a network with a low data rate. In a network of nodes creating 512 b/s or one 512 byte segment every eight seconds each. With a total network rate of 51.2 kb/s, the expected queue length is short. The rationale for simulating such a network is to measure the average delay. Based on prior knowledge of throughput in chains it is known that increasing number of hops reduces the per node throughput. Simulations are performed for 90 seconds.

Another realization made during simulation is how the hidden node problem can lead to misleading results in a grid of nodes. Figure 4.10 show the latency measured in a grid of 100 nodes communicating with a low rate and random source and destination. The dashed line is the average latency measured in a grid without an airborne relay. On the first inspection, one might think that a network performs quite well. Even for a maximum of two hops, it appears like the relay is able to offload the ground network. Meanwhile, what is the case is that packets transferred over the relay are lost. Lost segments counted on the right Y-axis show that for a maximum of two and three hops, most of the transferred segments are lost or stuck in the interface queue.

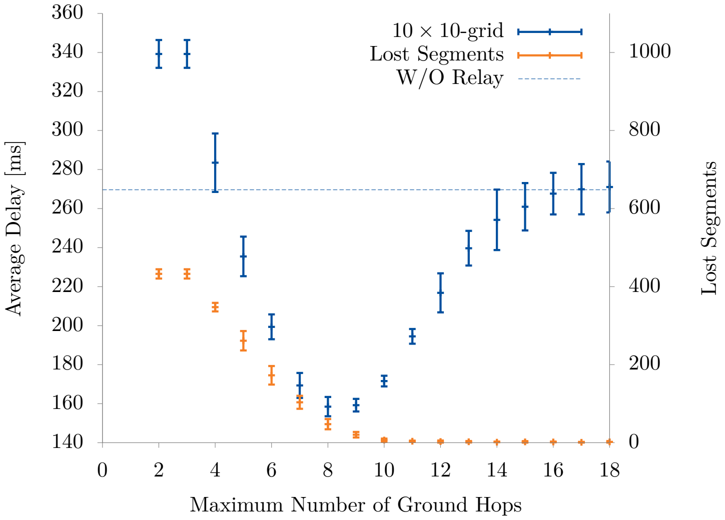


Figure 4.11: Delay of a slow rate of traffic sent from all nodes with random destination in a grid of nodes with RTS/CTS

With a higher number of nodes the probability of collisions increase. The number of received segments are not stable until MaxGroundHops is equal to thirteen or higher. This means that with only packets traveling thirteen hops or more using the relay, enough traffic is routed through the ground nodes that retransmission in the MAC layer is enough to avoid packet loss.

Figure 4.11 shows how the delay evolves with RTS/CTS activated. Like in the previous chart, the dashed line serves as the baseline measured as the average delay without a relay. The network still suffers from quite a lot of lost packets when the relay is under heavy use. Compared to the 1000 lost segments, this is a great step in the right direction regarding robustness. At a maximum of ten hops, the segmentation loss is equal to the loss at thirteen without the four-way handshake. Still, the reduction in latency is not that apparent in the chart. The average latency in the first chart, without handshake, is in the order of 192 ms (186, 198) and the second 172 ms (169, 174), or a reduction of 10.42 %.

The chart in Figure 4.12 show how increasing the number of nodes result in higher delay because of the longer average path. Notice how a grid of 36 nodes with a MaxGroundHops equal to 5 perform in the order of a network of 16 without a relay. Another factor present in the chart is how the optimal number of hops seems to change. In a 4 × 4-grid, a maximum of six or more hops is of no use since the longest shortest path route is six hops long.

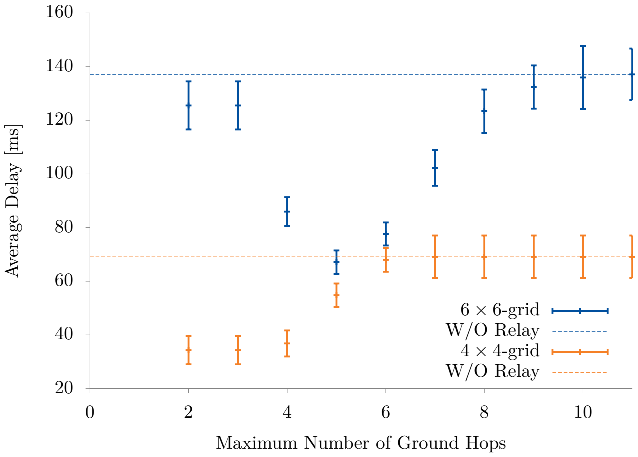


Figure 4.12: Delay of a slow rate of traffic sent from all nodes with random destination in a grid of nodes with RTS/CTS

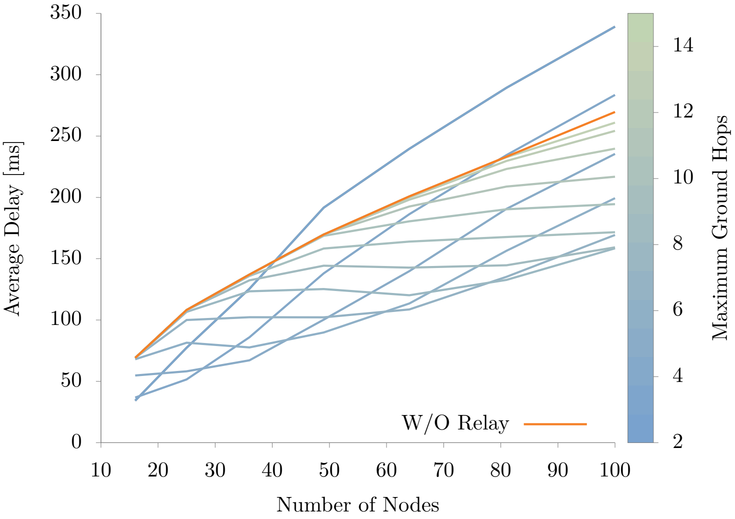


Figure 4.13: Comparing latency of different max hop attributes

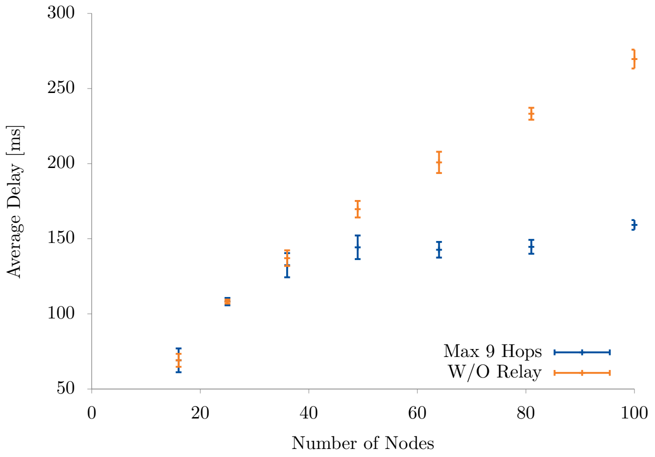


Figure 4.14: Comparing the latency of a normal grid with with one using maximum 9 hops

If a 6 × 6-grid can perform like a 4 × 4-grid when aided by a relay, does this imply that networks with an airborne relay scale better that grids without? Or more precisely, is the latency of a relay network lower than a normal network in more than just a single specified number of ground nodes? Figure 4.13 certainly appear more promising. Here, the orange lines show the average latency of a network with increasing number of nodes without a relay. The blue-green lines show different values of MaxGroundHops ranging between two and fifteen mapped to the scale on the right hand side of the chart. The majority of values lay bellow the orange baseline up to 100 nodes. As a result of the optimization problem, the values that perform best with few number of nodes is not the number who gives the lowest latency with more nodes and the other way around. At the same time, few of the lines perform worse than the baseline.

As an example of a max k-hop value suitable up to 100 nodes, Figure 4.14 has a chart that compares the latency in a regular grid with a network using a MaxGroundHop equal to nine hops. Figure 4.15 shows what happens when the number of nodes increases over 100 nodes. The Y-axis displays the ratio of lost segments compared to the number of created packets. Thus, the results suggest that within certain bounds a grid with a relay has a lower latency than one without over a set of different grids.

The next natural question to ask would be: At what cost? The relay network is equipped with twice as many interfaces as the one without. How would this evolve if both interfaces were used for ground communication? With two interfaces set to use orthogonal channels the expected capacity of the network would be doubled. Still,

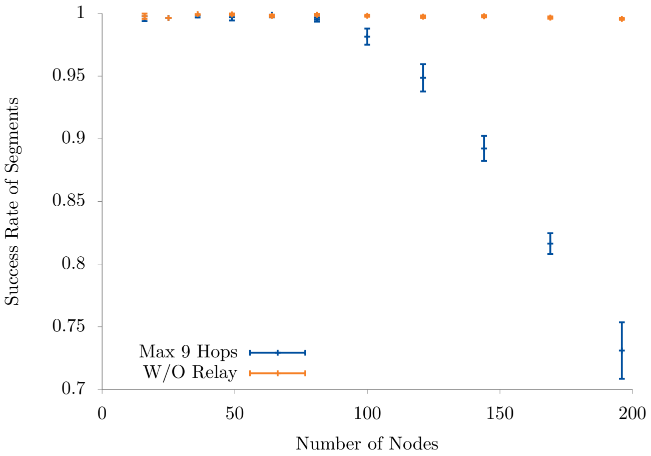


Figure 4.15: Comparing the latency of a normal grid with with one using maximum 9 hops

low traffic rates imply few packets in queue and a low delay contribution.

Assuming that each node can receive and transmit packets on both interfaces and that packets are buffered in a single queue, one can expect some reduction of nodal delay. It is more likely that one of the interfaces is in an idle state than in situations with only one interface. The transmission delay would however be unchanged. Thus, doubling they ground interfaces allows concurrent transmissions and higher throughput, but an airborne node can deliver lower latency at low rates.

Chapter 5 Discussion

Through a study of existing research and simulation, this thesis has surveyed limits in wireless MANET communication to find potential capacity gains by a deployment of an airborne node. Much of the literature evaluated as part of the thesis, is centered around general evaluations of MANET or specific technologies such as IEEE 802.11 in scenarios with nodes placed in a single plane. Based on the author's understanding of prior work and through simulation this thesis has contributed by evaluated scenarios where a relay gives a benefit and situations where it might not be able to contribute.

The presented results are only valid within the assumptions presented, and might not be directly relevant for real deployments of MANET. Simulations are performed under a strict placement of nodes. All simulations of networks who forward over multiple hops, nodes are placed at an optimal distance equal to the range of the network interfaces. In real networks placement is likely to follow a more irregular pattern. This wastes potential concurrent transmissions and thus lowers the capacity.

Further, the thesis uses a rather simple model for propagation loss. More realistic models for attenuation exist. Models that better reflect how the radiated power fades over a distance. On the other hand, the model used here assumes an immediate drop in radiated power. Such an approach is not suited for simulations where nodes have a more irregular placement. One example of where the current model does not fit well, is when multiple nodes are within range. In the event that both nodes transmit at the same time, it is not given that the collision results in a lost packet. When nodes are placed within receivable range, but at different distances, it is possible that the received power from the nearest node is strong enough to still be readable. At the same time, multiple nodes outside of interfering range might together raise the overall noise floor enough to render transmissions within normal range unreadable. Noise is another element that the thesis has abstracted out. All simulations assume a completely noise free environment. In reality noise affects reception, and might be changing inside the area of operation.

For large operations, it might not be realistic that all nodes stay connected with the relay. The developed propagation model gives all nodes equal communication condition with the relay with the relay always placed in a stationary position. Local differences such as obstructions as well as the relay's position would lead to a changing probability for correct reception. The modeled node is always placed at the center of the network, however for a real network, both the ground nodes and the relay would have to move. Reception can even be affected by the antenna polarity. If the platform of the relay uses wings to fly, the angle of the platform needs to change in order to steer, without advanced techniques for antenna pointing or beaming, the angle can affect the received power.

The placement is another limitation in the current approach. All simulation situations where a relay is active it is placed in the center. Another approach worth considering is how a relay could be placed to help out parts of a network in particular need. A relay could increase the capacity of parts of a network in particular need. This would reduce the potential nodes contending for the relay's resources from the whole network to only a subset.

A difficulty often stressed in research based on simulation, is its credibility. In [36] Kurkowski et al. finds that less than 15 % of the surveyed papers are reputable. If the results from a simulation are not credible or give little confidence, the simulation has little value. The initial validation of the simulation environment and the specially made components consists of tests using basic, deterministic, parameters and comparing the results with the expected results. The obvious shortcoming of this approach is that unexpected results might lead to changes to a model when in fact the results come from real and important effects.

Still, the issue regarding how one can be sure that a model is complete enough stands. Is it possible to prove that the simulator provides accurate and relevant samples in a given scenario? Testing and formal proofs are well-known methods that try to answer this question. By testing a simulation implementation in situations where results from the real system exist one can show that the simulator gives correct results under a particular circumstance. Meanwhile, it might not be practical to test all possible cases of a system. Formal proofs, on the other hand, like mathematical modeling, require a precise study of system behavior with a high degree of complexity.

The measurements on a single cell and a chain of node matches results from prior research and the calculated theoretical upper bound for efficiency. Some variation from [3] where identified when it comes to capacity in one cell of nodes. At the same time, [2] seems to support the findings. The performed simulations uses a fixed rate, that disables rate-adaption. Networks who deploy multiple rates with automatic rate adaption might suffer more from contention than networks with a fixed low rate.

In the first chapter, two questions are raised. The first question asks how the throughput capacity of two classes of traffic scale with increasing number of nodes and the distribution of traffic. Simulation results from a grid of nodes indicates a lower delay can be achieved with a relay in a network that forwards with a max k-hop rule. This is true within an upper band of nodes. No fixed value for the best number of hops is found. Instead, the results indicate that increasing number of relay users leads to a need for max-hop adjustments. To some extent, simulation results support the hypothesis, at least for latency. Another important factor to consider is how, too many contending nodes may lead to high drop rates — which emphasize the importance of choosing an appropriate value. Future work might lead to algorithms for finding the optimal k, and be able to deliver lower latency over a wider range of network sizes.

The next question in the introduction concern how the capacity in a relay-aided network compares to a pure ad hoc networks. Experience from simulations on chains of nodes with one source of traffic shows that the total throughput of marked traffic traveling over a relay stays constant where ground nodes experience a loss. Simulations with multiple sources, however, indicate that the hidden node problem and the lack of spatial reuse, result in lower throughput over the relay than along the ground.

If the networks are placed on the same channel; the results imply that the relay will be prevented access to the medium. In normal operation, wireless network nodes contend under equal terms. Meanwhile, with multiple transmitting stations outside of each other's range, the relay will suffer from an unfair competition. This is true both for a grid of nodes and a chain. The unfair access competition implies that relay nodes should be placed on a separate channel. Time slots could be assigned to the ground paths and the relay path and effectively create two channels within the same frequency spectrum. This would require a new MAC implementation and nodes to synchronize their time to be effective, which might not be impossible if equipped with a GPS.

On the other hand, 802.11 already supports a mechanism for channel reservation. Until now, all stations have requested access to the medium with a RTS-frame when the channel is idle. Meanwhile, there is nothing but policy stopping a station from transmitting a CTS addressed to itself. In fact, the CTS does not contain the transmitter address. In IEEE 802.11, this reservation of the channel is known as CTS-to-self. [21, S. 9.3.2.11] The mechanism is most commonly used by stations on a nonbasic modulation who may not be heard by all stations. For instance networks with a combination of 802.11b and 802.11g. Both standards use the same channels, but 802.11b stations cannot interpret 802.11g data frames sent at a higher rate. A station who need to reserve the channel, transmits a CTS with its own address set as the receiver address and a duration value that will protect the pending transmission and an ACK. The CTS will tell receiving stations to set their NAV values and wait for the duration specified.

For sparse ground networks with many stations it might be impossible or rare for the relay to enter a idle state. The relay can use a shorter DIFS or reduce its contention window to gain a higher probability of winning access to the channel. An even more radical approach could involve tearing down ongoing transmissions to get access. Because of spatial reuse, the relay might never get free access. By transmitting continuously for more than the maximal duration of a data frame, all ground nodes will end up in backoff and receives a CTS without colliding.

Strictly speaking, the relay node does not need the medium through the entire network to transmit. The only requirement for the relay to transmit correctly is that the receiver is idle. On the other hand, how would the relay know that the ground node received the frame if it is unable to receive an ACK? More importantly, when the relay is not the source of information, how would packets arrive there at all? For the relay to receive in a busy network of only one channel, all ground nodes in range of the relay have to be interrupted. The simplest solution seems to be a combination of CTS-to-self and a RTS/CTS-handshake. A relay can regularly signal to all ground nodes that it needs access by interrupting all ground nodes and transmits a CTS-to-self with a duration of zero. Terrestrial nodes can now interpret the CTS as an invite, wait one DIFS and contend for access with a random backoff. The node with the lowest backoff transmits a RTS, followed by a normal CTS from the relay and data transfer. Now, the airborne node can either request more frames by a new relay invite or transmit the newly arriving packet directly to the destination initiated by a RTS from the relay. The relay needs a shorter backoff period than the terrestrial nodes to be sure to catch the channel. If the airborne node does not re-invite, the channel will automatically be handed over to the ground nodes when their backoff period runs out.

For a sparse grid of nodes with a random traffic pattern, ad hoc networks on the ground give a higher throughput because the shorter range leads to concurrent access to the channel. A relay node with unlimited range is limited equal to the bounds found for nodes within one cell. When all nodes are within range, the total capacity appears constant. In other words per node-throughput scales according to $\frac{1}{N}$. Effectively it seems like a connected ad hoc network provides more capacity if the dedicated relay interface is used to double the ground capacity.

Chapter 6 Conclusion

The performance of wireless ad hoc networks has gained a lot of attention. By the use of an introduction to relevant technologies and prior work in the field of MANETs, the thesis have developed and documented appropriate simulation models in the Ns-3 network simulator. The thesis contributes by presenting a set of simulated scenarios using the produced modules and give further insight into situations and effects that might be relevant for real scenarios.

The throughput capacity found for ground nodes inside a single cell stays constant. This means that for an increasing number of nodes the per node throughput follows $O\left(\frac{1}{N}\right)$. For chains of nodes where all nodes are placed optimally the throughput for one source follows $O\left(\frac{1}{N-1}\right)$ until five nodes. For more than five nodes the throughput follow $O\left(\frac{1}{4}\right)$. Because chains can utilize concurrent transmissions, sets of nodes can further increase throughput. Ground nodes connected to an airborne relay shares one transmission domain and will thus not provide a higher throughput than what can be obtained from a chain of two nodes, $O\left(\frac{1}{2}\right)$. Or a per node throughput of $O\left(\frac{1}{2N}\right)$.

Simulations of networks where the relay and ground nodes share a common channel do not seem promising with respect to throughput. Assuming that the relay has a better coverage of the network, this will result in less spatial reuse, and thus lower the available throughput. With the relay placed in a central position with a CSMA/CA implementation, such scenarios will increase the time spent in backoff. As a result, networks in such scenarios will perform worse concerning throughput than systems without a relay. Even though, results imply a low throughput, simulation show that in low rate networks a relay can provide lower delay than a grid of ground stations which indicate that a relay node with the ability to interrupt ground stations can reduce the average latency.

References

[1] J. F. Kurose and K. W. Ross, *Computer networking — A top-down approach*. Pearson International, 4th ed., 2008. ISBN: 9780321513250.

[2] A. Duda, “Understanding the performance of 802.11 networks (invited paper),” in *PIMRC*, 2008.

[3] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” *MIT Labaratory for computer science*, pp. 61–69, 2001.

[4] P. Ramjee and L. Deneire, *From WPANs to personal networks, technologies and applications*, vol. 6. Arctech House, 2006. ISBN: 1580538266.

[5] R. Hartley, “Transmission of information,” *Bell System technical journal*, vol. 7, no. 3, pp. 535–563, 1928.

[6] C. Shannon, “The mathematical theory of communication,” *Urbana*, 1949.

[7] ISO/IEC JTC 1, “Information technology – Open Systems Interconnection,” ISO 7498-5.100.01, 1994.

[8] D. Barroso, “Spotify labs, SDN internet router — part 2.” labs.spotify.com/2016/01/27/sdn-internet-router-part-2. [Accessed: 2016-06-12].

[9] A. Nasipuri, R. Castañeda, and S. R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” *Mobile Networks and applications*, vol. 6, no. 4, pp. 339–349, 2001.

[10] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing.” RFC 3561 (Experimental), July 2003.

[11] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR).” RFC 3626 (Experimental), Oct. 2003.

[12] T. H. Kunz, “On the inadequacy of manet routing to efficiently use the wireless capacity,” in *WiMob’2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, vol. 3, pp. 109–116 Vol. 3, Aug 2005.

[13] N. Abramson, “The ALOHA system – another alternative for computer communication,” *Fall Joint Computer Conference*, 1970.

[14] C. K. Toh, *Ad hoc mobile wireless networks: protocols and systems.* Pearson Education, 2001. ISBN: 9780130078179.

[15] C. L. Fullmer and J. J. Garcia-Luna-Aceves, “Solutions to hidden terminal problems in wireless networks,” in *ACM SIGCOMM*, 1997.

[16] V. B. Iversen, “Handbook in teletraffic engineering,” tech. rep., ITC/ITU-D, 2005.

[17] L. Kleinrock, *Queueing systems, Theory*, vol. 1. Wiley-interscience, 1975.

[18] M. Grossglauser and D. Tse, “Mobility increases the capacity of ad-hoc wireless networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1360–1369, IEEE, 2001.

[19] C.-M. Cheng, P.-H. Hsiao, H. T. Kung, and D. Vlah, “Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm,” in *IEEE Wireless Communications & Networking Conference (WCNC)*, Mar. 2007.

[20] “Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-1997*, 1997.

[21] “Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2012*, 2012.

[22] P. Gupta and P. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[23] B. Han and G. Simon, “Capacity of wireless ad hoc networks, a survey,” tech. rep., ENST Bretagne, 2007.

[24] M. Pidd and R. B. Castro, “Hierarchical modular modelling in discrete simulation,” in *Simulation Conference Proceedings, 1998. Winter*, vol. 1, pp. 383–389, IEEE, 1998.

[25] “What is ns-3.” www.nsnam.org/overview/what-is-ns-3, 2016. GNU GPLv2. [Accessed: 2016-05-07].

[26] P. L’Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton, “An object-oriented random-number package with may long streams ans substreams,” *Operations Research*, vol. 50, pp. 1073–1075, 2001.

[27] University of Washington, “ns3, documentation.” www.nsnam.org/documentation. [Accessed: 13.06.2016].

[28] M. Lacage and T. R. Henderson, “Yet another network simulator,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, ACM, 2006.

[29] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[30] L. F. Perrone and Y. Yuan, “Modeling and simulation best practices for wireless ad hoc networks,” in *Simulation Conference, 2003. Proceedings of the 2003 Winter*, vol. 1, pp. 685–693, IEEE, 2003.

[31] L. Landmark, K. Øvsthus, and Ø. Kure, “Routing trade-offs in sparse and mobile heterogeneous multi-radio ad hoc networks,” in *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM*, pp. 2229–2236, IEEE, 2010.

[32] M. Allman, V. Paxson, and E. Blanton, “TCP Congestion Control.” RFC 5681 (Draft Standard), Sept. 2009.

[33] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” vol. 17, pp. 1–14, Elsevier, 1989.

[34] S. M. Mirhosseini and F. Torgheh, *ADHOCTCP: Improving TCP Performance in Ad Hoc Networks, Mobile ad hoc networks: Protocol design.* INTECH Open Access Publisher, 2011. ISBN: 9789533074023.

[35] E. Larsen, “TCP in MANETs — challenges and solutions,” tech. rep., Norwegian Defence Research establishment (FFI), 2012. 2012/01289.

[36] S. Kurkowski, T. Camp, and M. Colagrosso, “Manet simulation studies: the incredibles,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, 2005.