UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

MmWave Beam Control for Geocasting in Vehicular Networks

Thijs Havinga

MSc Thesis

Electrical Engineering

July 2021

Graduation committee:
Chairperson: prof.dr.ir. G.J. Heijenk
Daily supervisor: dr. S. Bayhan
External member: dr. C. Stegehuis

Design and Analysis of
Communication Systems
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Contents

List of Acronyms

CAM	Cooperative Awareness Message
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear-To-Send
C-V2X	Cellular V2X
DCST	Directional Constrained Steiner Tree
DENM	Distributed Environmental Notification Message
DMG	Directional Multi-Gigabit
FIFO	First In, First Out
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MILP	Mixed-Integer Linear Program
MPR	Multipoint Relay
MPRD	Multipoint Relay Destination
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing Protocol
PHY	Physical layer
RSU	Road-side Unit
RTPR	Residual Time Per Receiver
RTS	Request-To-Send
SC	Single Carrier
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

List of Symbols

Symbol	Description	Unit
$\mathcal{N}$	Set of nodes $\{0, \ldots, n, \ldots, N-1\}$ equipped with a sub-6GHz and mmWave radio interface.	
T, R	A node in $\mathcal{N}$ that transmits or receives a mmWave frame, respectively.	
$\mathcal{M}$	Set of generated mmWave messages $\{m_0, \ldots, m_j, \ldots, m_{M-1}\}$.	
n_{m_j}	Node in $\mathcal{N}$ that generated mmWave message m_j (also called origin node).	
$\mathcal{R}_{m_j}$	Set of intended receivers of message m_j.	
S	Size of a mmWave message.	Mbits
O	Timeout of a mmWave message.	s
$t_{m_j}^{gen}$, $t_{m_j}^{out}$	Time at which message m_j was generated and its timeout, respectively.	s
$t_{\min}$, $t_{\max}$	Minimum and maximum time considered for the scenario.	s
$PL_{T\to R}$	Pathloss via mmWave link from transmitter T to receiver R.	dB
$d_{T\to R}$	Distance from the antenna of transmitter T to the one of receiver R.	m
A	Pathloss exponent, depending on the number of obstructions on a link.	
C	Constant in path loss model, depending on the number of obstructions on a link.	dB
$\mathcal{W}$, w_{ml}, $w_{\min}$	Set of half-power antenna beamwidths w, main-lobe beamwidth and minimum antenna beamwidth, respectively.	$^\circ$
a, α	Pointing direction of an antenna and the observed angle relative to this direction, respectively.	$^\circ$
$\mathcal{A}$	Set of possible transmission direction angles for a node.	
G, G_0, G_{sl}	Directional antenna gain, maximum antenna gain and side-lobe gain, respectively.	dBi
P_T	Fixed transmission power of a transmitter T.	dBm
$P_{T\to R}$	Power of a signal as received by receiver R transmitted by T.	dBm
P_{sens}	Minimum power required for a receiver to be able to decode a transmission, also called receiver sensitivity.	dBm
r, $\mathcal{D}$	Data rate for a mmWave transmission and the set of possible data rates, respectively.	Mbps
$\mathcal{T}$	Transmission tree.	
$D_{\mathcal{T}}(n)$	Delay of a node n in a transmission tree $\mathcal{T}$.	s
$\mathcal{G}$	Graph consisting of vertices (nodes) and edges (links).	
$\mathcal{V}$	Nodes that are already visited while setting up a transmission tree.	
$\mathcal{U}$	Intended receivers still to be visited while building a transmission tree.	
$\mathcal{E}$	Set of edges in a graph representing links between nodes.	
$E(T \to \mathcal{R})$	Edge representing a link between T and $\mathcal{R}$.	
$E_C(T \to \mathcal{R})$	Cost of an edge between T and $\mathcal{R}$.	
$\mathcal{P}(T \to R)$	Shortest path between transmitter T and receiver R.	
$\mathcal{P}_C(T \to R)$	Cost of shortest path between transmitter T and receiver R.	

Preface

This thesis is a result of my final project for the master Electrical Engineering at the University of Twente, which I carried out at the research group Design and Analysis of Communication Systems (DACS). Due to the regulations against COVID-19, I carried out the work fully from home. This is quite a different setting from what I was used to, but since I really liked the subject, I was motivated and enjoyed doing the research nevertheless. Though, I am grateful that the final presentation can be given on campus.

Next, I would like to thank Geert Heijenk and Suzan Bayhan for their supervision. In the course of the project, I received valuable feedback via e-mail and during the weekly online meetings we had. The meetings were quite lengthy from time to time, but always productive. Furthermore, I would like to thank Clara Stegehuis for giving feedback on the midterm presentation and draft version of this thesis. Finally, I would like to thank my family and friends who always supported me and made working from home more pleasant.

Thijs Havinga
Enschede, July 2021

Abstract

Due to the increasing amount of data generated by sensors in modern cars, the need for high data rate links in vehicular networking rises. A promising way to achieve this is using mmWave communications, although beamforming is needed to overcome the high propagation losses at these frequencies. Relaying might be needed to cover larger distances for the delivery of a message in a specific geographical area, called geocasting. Reaching multiple receivers at once (multicasting) can be achieved by using a wider antenna beamwidth, which comes at the cost of transmission range, while spatial sharing can be exploited using narrow beams. This thesis investigates if using multicasts is beneficial for the routing and scheduling of mmWave geocasts that need to be delivered before a timeout. A system with multiple data rates is considered, for which the optimal solution is sought by modeling it as a mixed-integer linear program. Furthermore, a heuristic algorithm is created that, as a first step, finds the links and their quality based on position information. A reduced graph representing the nodes and links in the system that are likely needed to reach all intended receivers is generated as a second step. Next, a transmission tree that specifies the links that should be used and the order in which transmissions are scheduled is found. Several methods to include multicast links are evaluated, of which some consistently outperform the unicast-only method. Using multicasts is especially advantageous in scenarios with multiple highway lanes. However, there is still a performance gap for the heuristic algorithm as compared to the optimal solution. Finally, the heuristic approach is transformed into a distributed algorithm, which can be controlled via beacons sent on an additional sub-6GHz band. When the mmWave schedule is congested due to multiple concurrent messages, the benefit of multicasts becomes even more clear.

Chapter 1

Introduction

Communication between vehicles promises to be an effective way to improve traffic safety, road use efficiency, to reduce fuel consumption and enhance driving comfort [1]. Nowadays, vehicles become more and more equipped with various sensors that perceive the environment. These sensors create signals either to warn the driver or even to let the vehicle act on it, as a first step towards autonomous driving. However, in a dynamic and crowded environment, a vehicle may not be able to obtain all the information it needs in time. Letting vehicles cooperate by sharing information using wireless communication has high potential on filling this gap.

Prior work on vehicular communications mostly considers the sub-6GHz frequency bands. However, it is expected that high amount of data from (camera) sensors will require higher data rates than can be achieved at these frequencies [2]. To this end, mmWave communications seems a suitable candidate. Nonetheless, the downside of this technology is its severe path loss and its sensitivity to blockage, resulting in very limited transmission range. This can be partly overcome by the use of beamforming, meaning to let transmitter and receiver point their antennas towards each other using a narrow beam. Yet, the process of beamforming leads to significant overhead in a dynamic environment, because it takes several training frames to obtain the channel quality, as explained in [3]. The channel quality is needed to determine which antenna sectors should be used, how these should be pointed and what data rate can be achieved. Using position information acquired via sub-6GHz communication is proposed as a solution to this training challenge, as presented in [2] and [4]. From the position information, the link quality can be estimated based on antenna and propagation models, such that the right links can be chosen.

An important concept in vehicular networking is geocasting, which means delivering a message in a certain geographical area. The route taken from the origin to all the intended receivers in the destination area, possibly via relaying nodes, greatly determines the performance of the communication. A possible route for an example situation is given in Figure 1.1. Other routes, e.g. that skip more relaying nodes, might result into reaching the intended receivers with less delay.

Various geographic routing protocols exist in the literature, but those are usually not specific to mmWave communications. Yet, its distinct nature as compared to sub-6GHz wireless

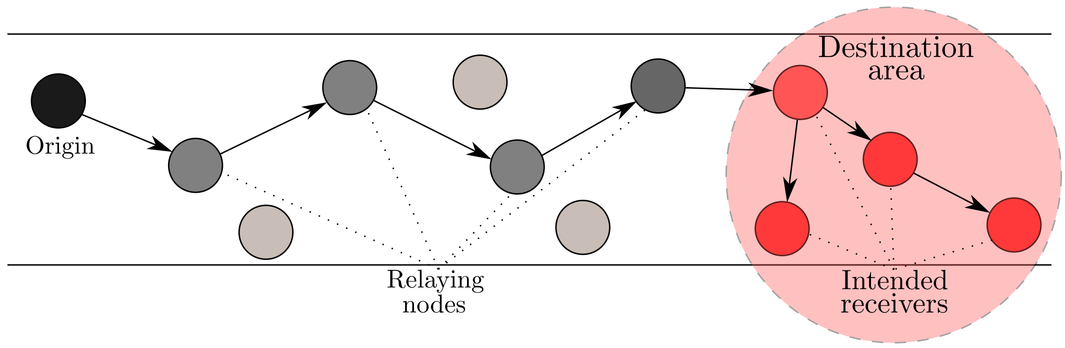


Figure 1.1: Route of a geocast towards a destination area via relaying nodes.

technology poses various challenges and opportunities that should be exploited to make full advantage of it. Firstly, spatial sharing, meaning scheduling multiple concurrent transmissions between different pairs of nodes, can be realized more often. Namely, transmissions using beamforming suffer less from interference, since the signals are not transmitted omnidirectionally. This is shown in Figure 1.2. Furthermore, using wider antenna beams allows for reaching multiple receivers at the same time (referred to as multicast), but it limits the transmission range [2]. Another aspect is introduced to this trade-off by using various data rates, which affects the transmission range as well. The combined effect of the latter two is shown in Figure 1.3.

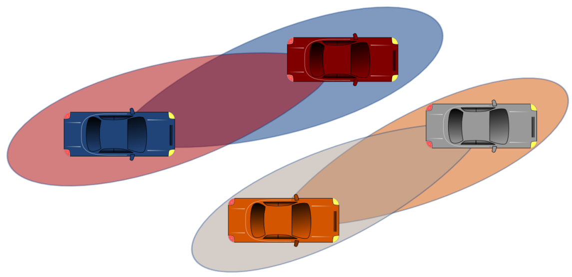


Figure 1.2: Directional antennas allow for spatial sharing.

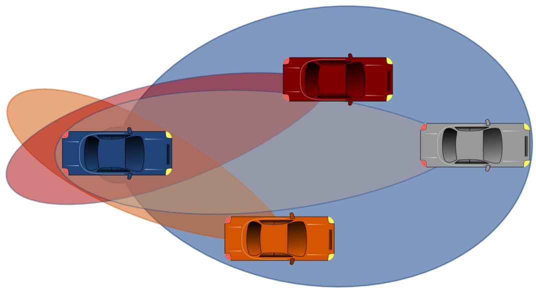


Figure 1.3: Using a wider beamwidth and lower data rate allows for reaching multiple receivers at once.

Geocasts in vehicular networks can be used to convey information such that the receivers can anticipate on upcoming situations that might occur. For example, a vehicle may inform other vehicles that it executes an emergency break, or that it wants to merge into a certain lane. If these messages are delivered when the vehicles can no longer act on it, the content is not useful anymore. Therefore, it is generally important that the geocast messages are delivered before a certain timeout. The aforementioned aspects of spatial sharing and using various beamwidths and data rates will be considered for the dissemination of a geocast, such that as much intended receivers as possible are reached in time. Furthermore, when multiple geocasts containing different messages should be delivered simultaneously, this should be taken into account by the system as a whole. Periodic transmissions (beacons) on the sub-6GHz communication layer that are, amongst others, used to distribute position information, can be used to control the mmWave transmissions as well [2]. Combining these two layers promises to be an efficient method to realize data dissemination in vehicular networks at high rates.

We formulate the following research questions that will be answered in this thesis:

1. *To what extent can mmWave geocasting benefit from using multicasts while considering multiple beamwidths and data rates, relaying and spatial sharing?*
2. *How can a mmWave geocast be efficiently routed and scheduled using a lightweight algorithm?*
3. *How can multiple mmWave geocasts be scheduled in a distributed manner using sub-6GHz beacons?*

The approach taken in this thesis is as follows. Question 1 will be answered by formulating the routing and scheduling problem as a Mixed-Integer Linear Program (MILP), which is solved with a mathematical optimizer in order to obtain an optimal solution. This gives insight into the theoretical performance of the system, either when only unicasts are allowed or when multicasts may also be used. Next, a heuristic algorithm that solves the problem with a lower computational complexity is created, which answers question 2. Lastly, to answer question 3, a procedure for scheduling multiple mmWave geocasts using sub-6GHz beacons is presented, providing a basis for a realistic implementation.

The remainder of this thesis is organized as follows. Chapter 2 presents the background and relevant literature on this subject and lists how this work deviates from it. In Chapter 3, an overview of the system is given. Next, the MILP will be formulated in Chapter 4. A breakdown of the problem and the steps needed to create a heuristic algorithm for it are given in Chapter 5. These steps are then worked out in Chapters 6-8. For each step, an evaluation of the performance is given. Chapter 9 presents the overall results, in which the heuristic algorithm is compared with the optimal solution and the influence of using multicasts is evaluated. Then, Chapter 10 elaborates on the distributed procedure using sub-6GHz beacons, that allows for scheduling multiple concurrent messages. Lastly, Chapter 11 concludes the research and lists suggestions for future work.

Chapter 2

Background and related work

In this chapter, we will provide the background and research on vehicular communications in general. Furthermore, we will give an overview of specific work concerning geographic routing and mmWave communications. Lastly, we will present the new research directions that will be addressed in this thesis.

2.1 Vehicular communications

Already in the early 2000s, researchers started to work on standards for vehicular networking in order to improve traffic safety and efficiency, to reduce fuel consumption and to increase driving comfort [1]. The idea is to create Vehicle-to-Vehicle (V2V), as well as Vehicle-to-Infrastructure (V2I) and even Vehicle-to-Pedestrian (V2P) communications, essentially obtaining Vehicle-to-Everything (V2X) communications. As a first stage, this can be used to warn drivers about road works or other hazardous situations. Later, when vehicles are equipped with several sensors, it is expected that they will exchange the created data and even negotiate about their driving behavior in order to obtain fully autonomous driving.

In pure V2V networks, there usually is a direct communication between the vehicles for which the link should be set up dynamically. Such networks are categorized as ad-hoc, meaning that there is no fixed infrastructure between the network elements. This poses various difficulties in order to ensure efficient communication. The first one is the dynamic and initially unknown link quality between nodes, because usually all links are wireless. The link quality therefore depends on the propagation conditions between nodes. The propagation is influenced by the distance between nodes, but also by objects in between them and other physical aspects, such as the weather. Especially in vehicular networks, dynamic link quality is of great concern, as the vehicles are highly mobile, thereby influencing the propagation conditions at all times. It is therefore required to monitor the link quality regularly.

The second problem is the existence of redundant links on the one hand, and situations in which links are scarce on the other hand. In ad-hoc networks usually every node can forward data, such that choosing the best route for the dissemination of a message is not trivial. Many ad-hoc routing protocols exist, of which several ones designed specifically for vehicular networking will be discussed later on.

Another issue is the access to and sharing of the medium. If there is no control, nodes can transmit simultaneously, resulting in interference. This will likely cause a collision of transmissions if the nodes are close to each other. Making the best use of the medium, while minimizing interference is one of the challenges of wireless networks.

One of the technologies for vehicular networking that tries to tackle these problems is based on the IEEE 802.11 standard, commonly known as Wi-Fi. The United States Federal Communications Commission (FCC) names this type of vehicular communication Dedicated Short-range Communication (DSRC) [5], whereas the European Strategy on Cooperative Intelligent Transportation Systems (C-ITS) calls it ITS-G5 [6], referring to the 5 GHz frequency band. Another promising technology, Cellular V2X (C-V2X) is mainly developed by the 3rd Generation Partnership Project (3GPP) and initially uses Long-Term Evolution (LTE) as underlying technology [7].
With the aforementioned technologies, vehicle state information, such as position and speed, are regularly distributed using Cooperative Awareness Messages (CAMs) [8]. On the other hand, safety information that needs to be distributed to a specific geographical location, triggered by an application, are sent in Distributed Environmental Notification Messages (DENMs) [9]. These ways of conveying messages are called beaconing[1] and geocasting, respectively.
Since the communication between vehicles is often meant to influence or even control their driving behavior, it forms a real-time application with hard time constraints. A specific version of a geocast, which needs to be delivered within a specific time is called an abiding geocast [10]. A geocast can be specified only by the geographical area in which it should arrive, but a specific node in the area of interest can be addressed as well. In the literature, there exists several protocols to ensure that messages reach the geographical area, which are described in the next section.

2.2 Geographic routing

In [11], several geographic routing protocols for vehicular networking are discussed. Their requirements differ from topology-based routing protocols in mobile ad-hoc networks. On the one hand, vehicles are assumed to be equipped with a GPS and memory storage and energy consumption are usually not a constraint. On the other hand, in the specific case of safety messages, delay constraints are very strict as they control a real-time system.
A popular non-delay tolerant protocol is called Greedy Perimeter Stateless Routing from [12]. The basic idea is to let a node forward a packet to a node that is geographically closer to the destination than itself, until the destination is reached. If the packet ends up at a node where no further progress can be made (local optimum), the node recovers from this by an adjusted algorithm.
In order to avoid ending up in a local optimum Spatially Aware Packet Routing, as presented in [13], uses topology information to set up a graph with nodes and links between them. It

[1] The terms CAM and beacon are used interchangeably.

then calculates the shortest path to the destination and selects the neighbor with the shortest path as next hop. Still, due to the changing topology, this does not guarantee that the packet will reach the destination, such that a recovery method is needed.
Several more sophisticated graph-based geographic routing protocols exist in the literature, which are designed to optimize for a specific metric or to be suited in a certain environments, e.g. urban or highway scenarios.

2.3 MmWave communications

For transmitting state information and safety messages, the data rates as supported by DSRC, ITS-G5 and C-V2X seem sufficient. However, it is generally believed that sensor data and planned trajectory information require increasing bandwidth. According to [14], the maximum data rate of the sub-6GHz-based standard IEEE 802.11p is in practice only 2-6 Mbps, whereas for the aforementioned applications at least 50 Mbps per vehicle is required, as explained in [2]. Therefore, research has been conducted on utilizing the millimeter wave (mmWave) spectrum for vehicular communication as well. Since this spectrum ranges from 30 to 300 GHz, it offers much larger bandwidth than the aforementioned technologies, which operate at the sub-6GHz band.
Adapted Wi-Fi-based standards supporting mmWave frequencies are IEEE 802.11ad, ay and bd [15], [16]. At the time of writing, the penultimate one has yet to be approved and the latter is still under development. For C-V2X, the use of 5G New Radio (NR) is considered by 3GPP in [17] and further worked out by 5G Communication Automotive Research and innovation (5GCAR) in [18].
A great downside of the use of mmWave frequencies is its high propagation loss, resulting in poor transmission range. This problem can be mitigated by beamforming, which will be addressed next.

Beamforming
Beamforming is the process of pointing the antennas of a transmitter and receiver towards each other using narrow beams. In this way, the transmit power is more concentrated towards a certain area, resulting in a larger transmission range as compared to omni-directional antennas. However, especially in vehicular environments, neighbor discovery, beamforming and scheduling at mmWave frequencies might result in significant overhead [2]. This becomes clear when investigating the beam training techniques proposed in the standards for mmWave communication.

Traditional beamforming
We consider the IEEE 802.11ad and ay standards for traditional beamforming here. IEEE 802.11bd plans to support both sub-6GHz and mmWave frequency bands. For the mmWave bands, it is proposed to upgrade the Physical layer (PHY) and lower Medium Access Control (MAC) layer to those of ad/ay [16].
IEEE 802.11ad and ay specify the use of multiple Directional Multi-Gigabit (DMG) beacon

frames in each antenna sector in order to discover nodes and perform beamforming. Receivers respond with sector sweeps towards the transmitter, including feedback for the best sector. If multiple nodes are in reach of the transmitter, they should respond in a contention-based period, leading to additional overhead. After the beam training, the data transfer interval starts, which might consist of a various number of contention based access periods and scheduled service periods. The latter should be reserved upfront by using a service period request in the DMG beacons. On the other hand, during contention based access periods, stations can compete for the medium via Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).
For 5G NR C-V2X, several beam scan schemes are still in consideration and might change depending on performance metrics, as described in [18]. Furthermore, this document proposes an approach in which beam scans are performed at first and once a link has been established, position information that is transmitted via this link is used for beamforming.

Position information

It becomes clear that the aforementioned method of IEEE 802.11ad/ay introduces significant overhead, especially when many antenna sectors are used and the node density is high. It has been shown that using vehicle position information can outperform traditional beamforming approaches of IEEE 802.11ad in terms of average network throughput, up to a mean position error of 3m in [4]. Instead of using sector level sweeps, antenna and propagation models are needed to estimate the received power. Position information should be received out-of-band, for example using sub-6GHz technology, in order to achieve overhead-free beamforming. Moreover, it has been shown in [19] that channel state information obtained via sub-6GHz shows correlation with that of mmWave and might even predict upcoming mmWave blockages, as described in [20]. The disadvantages of traditional beam training with highly mobile nodes and the potential of using position information via sub-6GHz frequency bands, makes this an interesting research topic.
The authors of [2] address the use of sub-6GHz V2V technologies for the scheduling of beamformed mmWave transmissions. Sub-6GHz communication is used as control plane in combination with mmWave technologies as data plane. A schedule for access to the mmWave channel is proposed using the status information from sub-6GHz beacons including location, speed, acceleration and heading direction. A mmWave transmitter includes a Request-To-Send (RTS) in a beacon to announce that it wants to send a message. Receivers can respond with a Clear-To-Send (CTS) including the time at which the transmission should start, which it bases on the mmWave schedule as planned up till now. Suggestions for future research are to exploit the possibility to schedule multiple receivers at the same time and relaying in order to reach vehicles at larger distances and to obtain spatial sharing. This forms the basis of the research presented in this thesis. Next, we summarize previous works related to routing and scheduling specific for directional antennas.

Directional routing and scheduling

There exist some studies on routing and scheduling with directional antennas in order to provide either multicast, relaying, spatial sharing, or a combination of these.

In [21] and [22], only multicast grouping is considered. Both describe a method in which for one transmitter the optimal sets of multicast receivers are determined. The authors of [21] propose an algorithm in which first the farthest receiver is selected as reference device. A beam pointing towards this receiver is considered, of which the beamwidth is iteratively incremented in order to maximize the product of number of receivers and data rate to cover those. This group is selected and then the process starts again by assigning a new reference device, until all receivers are included. In [22], only receivers for which a data rate can be achieved that is higher than a certain threshold are evaluated as a multicast receiver. Then for each of the antenna patterns they define, a weight is determined, which is regarded as the average per-frame transmission time for each client. Each time, the antenna pattern with the smallest weight is selected and the corresponding receivers are scheduled. This is repeated until all receivers are covered.

The authors of [23] address both multicast and relaying, a problem that they prove is NP-complete. In their algorithm, a node becomes a transmitter for each frame to be sent once it has received the data. Each transmitter creates a subset of receivers by first choosing the closest node and then adding receivers if the additional distance and angle needed is lower than a predefined threshold. Then a time-slotted scheme is created such that the maximum achievable rate for this frame is obtained. The algorithm performs better than using only multicast or relaying.

Both relaying and spatial sharing is considered in [24], but they consider unicasts only. Here, it is assumed that a Road-side Unit (RSU) knows the network topology and determines the transmitters, receivers and transmitting scheme. It does this by maximizing the number of transmissions in a time slot.

In [25], a heuristic algorithm is presented that offers the combination of multicast, relaying and spatial sharing, which is proven to be NP-hard. In this method, the closest transmitter is selected for each receiver. Then each transmitter determines to whom of the selected receivers it will send in the upcoming time slot by maximizing the sum throughput. The results show the importance of using relaying and spatial sharing.

A routing and scheduling protocol for multirate wireless networks with directional antennas is presented in [26]. They consider multicast, relaying, as well as spatial sharing. In order to determine which node should relay to which receiver set, they present the average broadcast time. This is the time that is needed to reach a certain set of receivers, divided by the number of receivers in that set. For scheduling, they use the notion of remaining broadcast time, which is defined as the longest delay that a node has to reach a receiver. Nodes with a high remaining broadcast time get priority in the schedule. It is shown that both the use of multiple datarates and directional antennas outperforms systems which do not include those in terms of transmission delay.

2.4 New research directions

This thesis considers a system which differs from the previous works on several points, as listed as follows.

- **Non-time-slotted system**
  The algorithms presented in [23]–[25] reason from a synchronized time-slotted system, whereas the IEEE 802.11 standards use CSMA/CA, which is non-time-slotted. Therefore, the problem here is addressed from a non-time-slotted perspective, meaning that transmissions can be scheduled at any point in time.

- **Selection of intended receivers**
  In the previous works, usually all nodes are considered to be intended receivers. As for geocasting the message should only be delivered in a certain area, not all nodes will need to receive the message. This gives rise to the problem of finding which nodes to use as relay nodes.

- **Multiple beamwidths, data rates and transmission direction angles**
  Like in [25], the system will jointly exploit multicasting, relaying and spatial sharing. However, in [25] a node can activate only active multiple antenna lobes to have a wider coverage at the cost of datarate. In the considered system vehicles may use different beamwidths, data rates and direction angles to determine the coverage of a transmission.

- **Realistic antenna model including obstacles**
  In [25], the antenna coverage is assumed to take the form of a sector of a circle. In this system, we consider a realistic antenna model, like in [23]. Furthermore, the influence of vehicles that form an obstruction for the signal will be taken into account in the path loss model that is used.

- **Multiple geocasts**
  [23] and [25] consider the scheduling of only one message (geocast). In vehicular networks, it is expected that multiple vehicles want to send a mmWave message. Hence, in the situation considered here, all vehicles may generate a mmWave message at different points in time.

As described, some of the points mentioned here are separately examined in previous works. To the best of our knowledge, there exists no work that considers all these points jointly.
Like in the previous works, the greatest simplification of the system is to consider static snapshots only. The essence of the problem does not change by this, but the practical implementation of an algorithm is much more complicated. Using vehicular data as speed, acceleration and heading, expected positions in the near future can be estimated, such that the routing and scheduling can be adapted upon this. However, this is out of the scope of this research.

Chapter 3

System overview

Hereafter, we will describe the elements of a scenario that considers the points as described in the previous chapter. In the course of the description, we will introduce symbols that are used in the rest of this thesis. Furthermore, we give the propagation and antenna models used to model the mmWave Physical layer (PHY). Besides, parameters that are used throughout the thesis are given.

3.1 Description

There are N static nodes capable of sending and receiving mmWave messages placed in a 2D-plane, which comprise the set $\mathcal{N} = \{0, 1, \ldots, N-1\}$. A transmitter $T \in \mathcal{N}$ is the specific node that sends a mmWave message m_j. We denote by $\mathcal{M} = \{m_0, \cdots, m_j, \cdots, m_{M-1}\}$ the set of mmWave messages generated during the considered time of a scenario from $t_{\min}$ to $t_{\max}$. A mmWave message m_j is generated at time $t_{m_j}^{gen}$, which is triggered by an application on a higher level by node $n_{m_j} \in \mathcal{N}$, called the origin node. The set of intended receivers $\mathcal{R}_{m_j}$ of message m_j are nodes other than the origin node, which are positioned in a specific geographical area, or are individually addressed. An intended receiver should receive the message before the timeout O, given by the transmitter. Per scenario, we will use a fixed timeout for all messages, which all have a fixed size S (in Mbits). Initially, only the origin node can send the mmWave message. After another node received this message, it can transmit it to others as well. We assume that the mmWave communication is half-duplex, meaning that nodes cannot transmit and receive at the same time. Furthermore, a node can only transmit or receive one message at a time. For an example scenario the input for the system, consisting of the positions of nodes in the 2D-plane and a timeline of message generation times and timeouts is given in Figure 3.1.
Next, the propagation and antenna models are discussed, from which it can be concluded whether a set of nodes can be reached by a transmitter given a beamwidth, data rate and transmission direction angle.

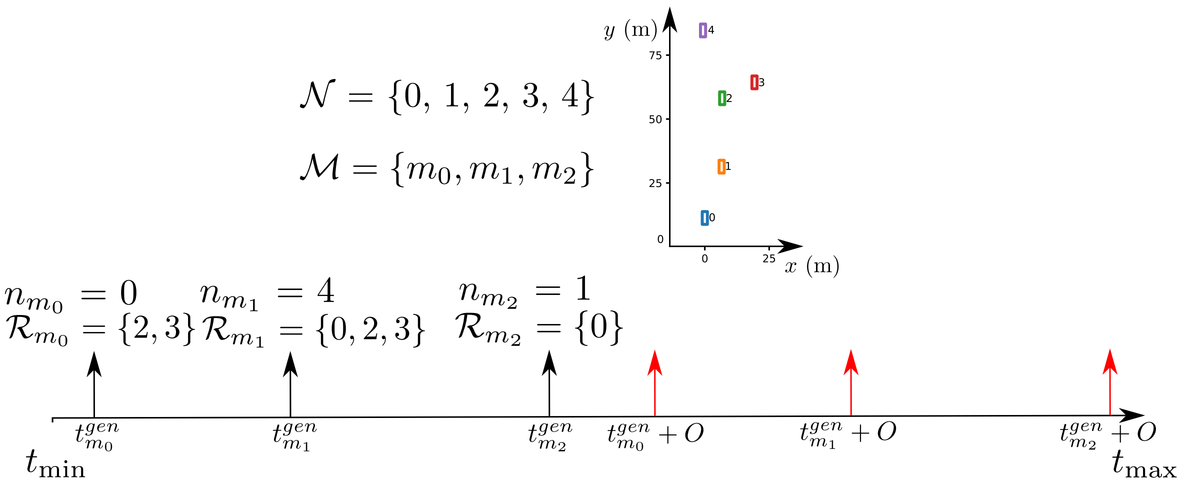


Figure 3.1: Input for the system for an example scenario.

3.2 Propagation and antenna models

In order to model the propagation of the mmWave communication, we will use an empirical path loss model derived for vehicular communication at 60 GHz from [27]. This model includes measurements under Line-Of-Sight (LOS) and Non-LOS (NLOS), when the link is obstructed by one or more vehicles. Other objects influencing the propagation, such as buildings, are not taken into account, except for a smooth surface underneath the vehicles. The path loss $PL_{T\to R}$ in dB between a transmitter T and receiver R is given by [27]:

$$PL_{T\to R}(d_{T\to R}) = A \cdot 10 \cdot \log_{10}(d_{T\to R}) + C + 15 \cdot \frac{d_{T\to R}}{1000}, \tag{3.1}$$

where $d_{T\to R}$ is the distance between the antenna of the transmitter and receiver (in m), A is a coefficient referred to as the path loss exponent and C is a constant.
Both A and C depend on the number of vehicles obstructing the link. To determine this number, we will use the number of nodes that are crossed when a straight line from the transmitter towards the receiver is outlined, using a fixed car width and length.
Each node can transmit with a fixed power P_T (in dBm), limited by the regulatory authority in a certain region. Both the transmitter and receiver use a beamforming antenna. The transmitter can choose to transmit using a beamwidth $w \in \mathcal{W}$ and a transmission direction angle $a \in \mathcal{A}$. The set of beamwidths is determined by the number of sectors of the antenna. Furthermore, the transmitter can choose between a discrete number of angles in the range $[0°, 360°)$.

For the calculation of the directional antenna gain, we will use a symmetrical antenna model using an average side-lobe level and Gaussian shape main-lobe from [28]. This model is used in the mmWave-based IEEE 802.15.3c standard and is considered simple, yet realistic. The directional antenna gain G of a node transmitting using half-power beamwidth w (in degrees), observed at an angle $\alpha \in [-180°, 180°)$ relative to the direction of transmission, is given by:

$$G(w,\alpha) = \begin{cases} G_0 - 3.01 \cdot (\frac{2\alpha}{w})^2, & 0 \le |\alpha| \le w_{ml}/2 \\ G_{sl}, & w_{ml}/2 \le |\alpha| \le 180° \end{cases} \tag{3.2a}$$

$$w_{ml} = 2.58 \cdot w \tag{3.2b}$$

$$G_0 = 20 \log_{10} \left(\frac{1.6162}{\sin(w/2)} \right) \tag{3.2c}$$

$$G_{sl} = -0.4111 \cdot \ln(w) - 10.579 \tag{3.2d}$$

where w_{ml} is the main lobe width (in degrees), which may range from 15° to 60°. G_0 is the maximum antenna gain and G_{sl} is the side lobe gain. A polar plot of the directional antenna gain for different beamwidths is given in Figure 3.2.

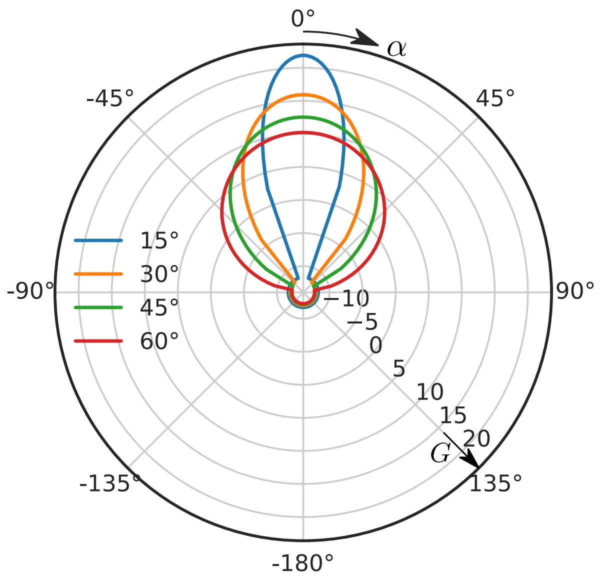


Figure 3.2: Polar plot of the directional antenna gain for different beamwidths, directed towards 0°.

If a node has been addressed as a receiver, it will point its antenna towards the transmitter using the narrowest possible beamwidth $w_{\min}$, resulting in an antenna gain equal to $G_0(w_{\min})$. The power from transmitter T using beamwidth w as received at an angle α at the receiver R is then given by:

$$P_{T\to R} = P_T + G(w,\alpha) + G_0(w_{\min}) - PL_{T\to R}(d_{T\to R}). \tag{3.3}$$

A mmWave transmission is possible if the power from transmitter to receiver is higher than a certain threshold, the receiver sensitivity P_{sens} (in dBm), such that the signal can still be decoded. The receiver sensitivity is defined by the Modulation and Coding Scheme (MCS) that is used, based on a maximum allowed Packet Error Ratio (PER) and the specified data rate r (in Mbps) for that MCS. The directional multi-gigabit PHY specification of the IEEE 802.11 standard [29] comprises of 31 different MCSs (19 for Single Carrier (SC) and 12 for Orthogonal Frequency Division Multiplexing (OFDM)), giving an equal number of distinct receiver sensitivity and data rate combinations. The available data rates r comprise the set

$\mathcal{D}$ and the corresponding receiver sensitivity is given by $P_{sens}(r)$. Unless otherwise specified, the parameters used in this paper are as given in Table 3.1. For the specified receiver sensitivities, 5 dB implementation loss and 10 dB noise factor are taken into account.

Table 3.1: Parameters for the system as used in this thesis.

Parameter	**Value(s)**	**Unit**
$\mathcal{A}$	{0, 1, 2, ..., 359}	°
$\mathcal{W}$	{15, 30, 45, 60, 360}	°
$r \in \mathcal{D}$; $P_{sens}(r)$	{693; -66, 866.25; -64, 1386; -63, 1732.5; -62, 2079; -60, 2772; -58, 3465; -56, 4158; -54, 4504.5; -53, 5197.5; -51, 6237; -49, 6756.75; -47} (Each MCS of OFDM.)	Mbps; dBm
P_T	10	dBm
#obstacles: A; C	0: 1.77; 70 1: 1.71; 78.6 2: 0.635; 115 >2: 0.362; 126	-; dB
S	300	Mbits
Car width	2	m
Car length	5	m

While it is not possible for a node to transmit or receive multiple messages at a time, concurrent transmissions between sets of nodes that do not overlap are possible. For the sake of simplicity, we will not take interference that might occur due to concurrent transmissions into account. Due to the directivity of the antennas, especially the receiving beam, the influence of interference will be limited in realistic scenarios. Moreover, as moving vehicles keep distance between each other, their antennas will not be very close to each other, such that it is unlikely that they pick up significant power from unwanted signals.

The complete system has now been described. Next, for any possible scenario we want to find the best route and schedule for mmWave transmissions, such that the most intended receivers are reached before the timeout, for each generated message. Before designing a practical and distributed algorithm for this, first the optimal solution is determined via linear programming in the next chapter.

Chapter 4

Optimal routing and scheduling via linear programming

In this chapter, an optimal solution is searched for the routing and scheduling of mmWave transmissions by modeling the system as a mixed-integer linear program (MILP). In this situation, every vehicle is aware of the location of all other nodes and the scheduling of all transmissions. Normally, information about this should be obtained via sub-6GHz communication, but the overhead introduced by this is left out of the program. Besides, the generation of mmWave geocasts and their corresponding intended receivers is known upfront. In this way, the solution is determined by viewing the problem from the perspective of an oracle, which looks back in time.

The objective is to maximize the number of intended receivers reached before the timeout, for all generated messages during the considered time. We will introduce the notion of frames, which are opportunities to send a mmWave transmission, denoted by $\mathcal{F} = \{\cdots, f_i, \dots\}$. For each frame that is used, the message that is included in the transmission and the start time, transmitter and receiver set of the transmission should be specified. These are the decision variables for the program.

We will model the system as described in the previous chapter. As explained, the input of the system is a topology of nodes with a list of mmWave messages generated during the considered time. Thus, for the generation time of the messages holds:

$$t_{\min} \leq t_{m_j}^{gen} \leq t_{\max} \quad \forall m_j. \tag{4.1}$$

For each message, the origin node and intended receivers are specified. The set of intended receivers $\mathcal{R}_{m_j}$ for a message m_j is not empty and will never include the origin node n_{m_j}, since it generated the message. This is given by:

$$\mathcal{R}_{m_j} \neq \emptyset \subseteq \mathcal{N} \setminus n_{m_j} \quad \forall m_j. \tag{4.2}$$

Whether a specific node n is an intended receiver for message m_j is specified by the binary variable $\rho_{m_j,n}$. Furthermore, whether node n is the origin node of message m_j is captured by the binary variable $o_{m_j,n}$. Only one node can be the origin node, as given by:

$$\sum_{n \in \mathcal{N}} o_{m_j,n} = 1 \quad \forall m_j. \tag{4.3}$$

First, we will pre-process the input scenario in order to obtain the rest of the parameters that are given as input to the MILP, which is explained hereafter.

4.1 Pre-processing input parameters

In order to find the optimal solution, we need to provide all possible receiver sets for all nodes as input to the program, such that any way of reaching the intended receivers, possibly via relaying, is considered. We will generate these receiver sets upfront, which are determined by taking the antenna and propagation models into account. For each node in the topology, an exhaustive search through all possible beamwidths, data rates and transmission direction angles is done. If the condition $P_{T \to R}(w, a) \geq P_{sens}(r)$ is met, receiver R is a reachable receiver for transmitter T using the combination with beamwidth w, data rate r and transmission direction angle a. The distinct receiver sets that can be reached via a combination are determined by the powerset (except the empty set) of all receivers for which the condition holds. For each distinct receiver set that can be reached with any of the combinations, only the combination with highest data rate is saved, in order to limit the solution space. This preserves optimality, since reaching the same set of receivers at a lower data rate will never lead to reaching more intended receivers before the timeout.

From the reasoning above follows that the same beam might be used to reach a different set of receivers. As an example, consider the scenario with three nodes in a linear topology as in Figure 4.1. The coverage drawn here also incorporates the maximum receiver antenna gain, $G_0(w_{\min})$.

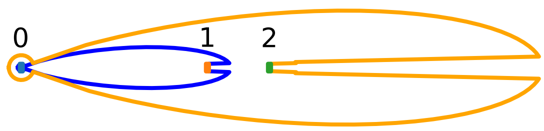


Figure 4.1: Three nodes in linear topology, showing the coverage of the beamwidth, data rate and angle combination for all sets of reachable receivers for node 0.

The sets of receivers that can be reached for transmitter 0 are $\{1\}$ and $\{1, 2\}$, with data rates of 3.5 and 0.7 Mbps, respectively. The sets in the powerset (except the empty set) of the reachable receivers of node 0 are: $\{1\}$, $\{2\}$ and $\{1, 2\}$. If node 0 chooses set $\{1\}$, it will use the combination that only covers node 1, because it has the highest data rate. On the other hand, if it chooses set $\{2\}$, the only option it has is to use the combination that also covers node 1. However, if it does not address node 1 for this frame, node 1 will not point its antenna and will still be available for other transmissions during that time.

If multiple combinations obtain the same data rate, the combination with narrowest beamwidth is selected. This is not strictly necessary, since interference is neglected, but intuitively this

seems a better choice, as it improves the directivity towards the receiver. Besides, the transmission direction angle is chosen arbitrarily if multiple are valid for a receiver set.
All distinct receiver sets $\mathcal{R}_l$ that are given as input to the program comprise the set $\overline{\mathcal{R}}$. Whether a receiver R is included in receiver set $\mathcal{R}_l$ is defined by the binary variable $x_{\mathcal{R}_l,R}$. If a node T is the transmitter of receiver set $\mathcal{R}_l$, the binary variable $y_{\mathcal{R}_l,T}$ takes a value of 1. The transmitter of a receiver set cannot be in the receiver set itself:

$$y_{\mathcal{R}_l,n} + x_{\mathcal{R}_l,n} \leq 1 \quad \forall \mathcal{R}_l, \forall n. \tag{4.4}$$

The duration $\delta^s_{\mathcal{R}_l}$ corresponding to the time it takes to reach a receiver set is determined by the data rate used. If for a receiver set $\mathcal{R}_l$ a data rate r can be achieved, this is given by:

$$\delta^s_{\mathcal{R}_l} = \frac{S}{r}, \tag{4.5}$$

where S is the fixed size of a mmWave message in Mbits.
The program decides on the number of frames to use to send mmWave messages. It can use the frames from the set $\mathcal{F}$, which comprises $F = M \cdot (N-1)$ frames, where M is the number of messages generated during the considered time and N is the number of nodes. This is defined such that every message may be sent to all nodes that are not the origin node. This bound does not lead to an additional constraint, as follows from the following reasoning. A receiver set cannot be empty, so at most one frame per intended receiver is needed. The maximum number of intended receivers of a message equals $N-1$. Furthermore, a node only needs to receive the message once. Thus, at most $N-1$ frames per message are needed to maximize the objective.
Now, all input parameters are determined and we will define the decision and helper variables of the program hereafter.

4.2 Program formulation

The decision variables are given as follows. If and only if the program decides to use frame f_i, the binary variable q_{f_i} takes the value 1. If the frame is used, the mmWave message m_j that is included in it should be selected. The binary value s_{f_i,m_j} specifies whether message m_j is included in frame f_i. Furthermore, a transmitter T and receiver set $\mathcal{R}_l$ are assigned to a frame that is used. The binary decision variables $u_{f_i,T}$ and $w_{f_i,\mathcal{R}_l}$ define whether transmitter T and receiver set $\mathcal{R}_l$ are assigned to frame f_i. Lastly, the program should choose the start time of a frame, as defined by decision variable $t^{start}_{f_i}$.
In this way, the program ultimately creates a transmission schedule to which all the nodes have to comply in order to obtain the optimal solution. A valid mmWave transmission schedule for the example scenario from Figure 3.1 is given in Figure 4.2. For each frame in the figure, it is specified which message is included, which node is the transmitter, at which time the frame starts, which beamwidth, data rate and transmission direction angle are used, and to which receiver set it is sent. The antenna beams are shown in the plots with positions of the nodes on the right.

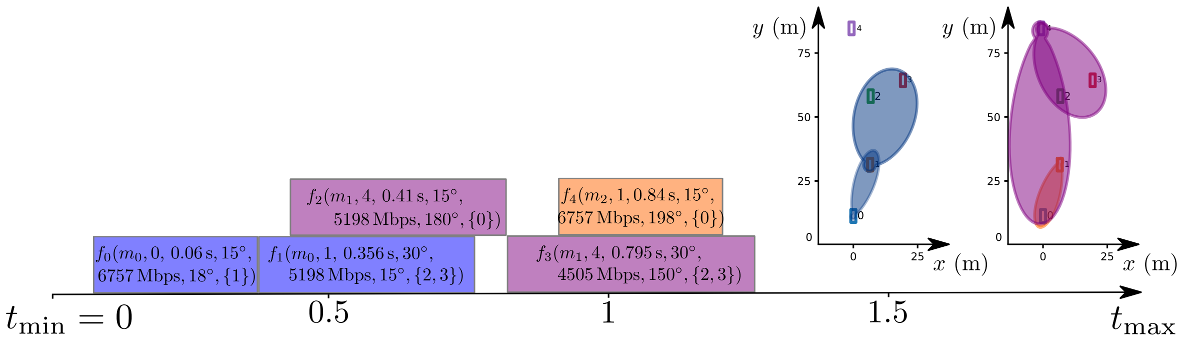


Figure 4.2: Output for the example scenario of Figure 3.1.

Next, we will formulate the constraints step by step. Some helper variables are needed to formulate these, which we will introduce along the way. The definition for the symbols of the parameters, decision variables and helper variables used in the MILP are summarized in Table 4.1. The program described hereafter is linear, because it can be written using linear constraints only. The program includes both binary and continuous variables. Therefore, the program is categorized as mixed-integer.

The constraints for the domain of the variables are defined as follows:

$$q_{f_i} \in \{0, 1\} \quad \forall f_i \tag{4.6}$$

$$s_{f_i,m_j} \in \{0, 1\} \quad \forall f_i, \forall m_j \tag{4.7}$$

$$u_{f_i,T} \in \{0, 1\} \quad \forall f_i, \forall T \tag{4.8}$$

$$w_{f_i,\mathcal{R}_l} \in \{0, 1\} \quad \forall f_i, \forall \mathcal{R}_l \tag{4.9}$$

$$t^{start}_{f_i} \in \mathbb{R}_{\geq 0} \quad \forall f_i \tag{4.10}$$

$$\delta^{f}_{f_i} \in \mathbb{R}_{\geq 0} \quad \forall f_i \tag{4.11}$$

$$v_{f_i,R} \in \{0, 1\} \quad \forall f_i, \forall R \tag{4.12}$$

$$\phi^{<}_{f_i,f_{i'}} \in \{0, 1\} \quad \forall f_i, \forall f_{i'} \tag{4.13}$$

$$s^{<}_{f_i,m_j} \in \{0, 1\} \quad \forall f_i, \forall m_j \tag{4.14}$$

$$\rho^{s}_{f_i,m_j,R} \in \{0, 1\} \quad \forall f_i, \forall m_j, \forall R \tag{4.15}$$

$$l_{f_i,f_{i'},m_j,n} \in \{0, 1\} \quad \forall f_i, \forall f_{i'}, \forall m_j, \forall n. \tag{4.16}$$

To start with, Constraints 4.17, 4.18 and 4.19 ensure that if a frame is used, there is exactly one message included, one transmitter assigned and one receiver set selected for it, respectively.

$$\sum_{m_j \in \mathcal{M}} s_{f_i,m_j} = q_{f_i} \quad \forall f_i \tag{4.17}$$

$$\sum_{T \in \mathcal{N}} u_{f_i,T} = q_{f_i} \quad \forall f_i \tag{4.18}$$

$$\sum_{\mathcal{R}_l \in \overline{\mathcal{R}}} w_{f_i,\mathcal{R}_l} = q_{f_i} \quad \forall f_i \tag{4.19}$$

Following, Constraint 4.20 specifies that if a receiver set is assigned to a frame, the transmitter corresponding to this set should send the frame. Furthermore, if a receiver is addressed in a

Table 4.1: Symbols and definitions for the parameters, decision and helper variables used in the MILP.

Symbol	**Definition**
Parameters	
$\mathcal{N} = \{\cdots, n/T/R, \ldots\}$	Nodes
$\mathcal{F} = \{\cdots, f_i, \ldots\}$	MmWave frames
$\mathcal{M} = \{\cdots, m_j, \ldots\}$	MmWave messages
$\overline{\mathcal{R}} = \{\cdots, \mathcal{R}_l, \ldots\}$	Set of all possible receiver sets
$t^{gen}_{m_j}$	generation time (in s) for message m_j
O	timeout of a mmWave message (in s)
$\rho_{m_j,n}$	$1 \iff n$ is intended receiver for m_j
$o_{m_j,n}$	$1 \iff n$ is origin node for m_j
$x_{\mathcal{R}_l,R}$	$1 \iff$ receiver R is in set $\mathcal{R}_l$
$y_{\mathcal{R}_l,T}$	$1 \iff T$ is transmitter for set $\mathcal{R}_l$
$\delta^s_{\mathcal{R}_l}$	duration (in s) for transmission to $\mathcal{R}_l$
Decision variables	
q_{f_i}	$1 \iff$ frame f_i is used
s_{f_i,m_j}	$1 \iff$ message m_j is included in f_i
$u_{f_i,T}$	$1 \iff$ transmitter T sends f_i
$w_{f_i,\mathcal{R}_l}$	$1 \iff$ receiver set $\mathcal{R}_l$ is included in f_i
$t^{start}_{f_i}$	start time of frame f_i
Helper variables	
$\delta^f_{f_i}$	duration (in s) of frame f_i
$v_{f_i,R}$	$1 \iff$ receiver R is addressed in f_i
$\phi^<_{f_i,f_{i'}}$	$1 \iff$ frame f_i ends before start of frame $f_{i'}$
$s^<_{f_i,m_j}$	$1 \iff$ frame f_i containing m_j ends before timeout
$\rho^s_{f_i,m_j,R}$	$1 \iff$ intended receiver R for message m_j is reached in time in f_i
$l_{f_i,f_{i'},m_j,n}$	$1 \iff$ node n received message m_j in frame $f_{i'}$ before f_i

frame, the specific set that contains this receiver should be chosen for the frame, as modeled by Constraint 4.21. Lastly, Constraint 4.22 ensures that the duration of a frame is equal to what is given for the receiver set that is selected for the frame.

$$2 \cdot w_{f_i,\mathcal{R}_l} \leq u_{f_i,T} + y_{\mathcal{R}_l,T} \quad \forall f_i, \forall \mathcal{R}_l, \forall T \tag{4.20}$$

$$2 \cdot v_{f_i,R} \leq w_{f_i,\mathcal{R}_l} + x_{\mathcal{R}_l,R} \quad \forall f_i, \forall \mathcal{R}_l, \forall R \tag{4.21}$$

$$\delta^f_{f_i} = \sum_{\mathcal{R}_l \in \overline{\mathcal{R}}} w_{f_i,\mathcal{R}_l} \cdot \delta^s_{\mathcal{R}_l} \quad \forall f_i \tag{4.22}$$

Next, if a message is included in a frame, the frame should start at a time equal to or greater than the generation time of that message, which is ensured by Constraint 4.23.

$$t^{start}_{f_i} \geq s_{f_i,m_j} \cdot t^{gen}_{m_j} \quad \forall f_i, \forall m_j \tag{4.23}$$

Constraint 4.24 enforces the helper variable $\phi^<_{f_i,f_{i'}}$ to be 1 if and only if frame $f_{i'}$ starts not earlier than the end of another frame f_i, i.e. these frames do not overlap.

$$\phi^<_{f_i,f_{i'}} = 1 \iff t^{start}_{f_i} + \delta^f_{f_i} \leq t^{start}_{f_{i'}} \quad \forall f_i, \forall f_{i'}, f_i \neq f_{i'} \tag{4.24}$$

A node cannot transmit and receive simultaneously, and a transmitter or receiver can be involved in a single transmission only. Therefore, Constraint 4.25 ensures that if a node is the transmitter of frame f_i and a receiver of frame $f_{i'}$, these should be sent at non-overlapping times. Furthermore, we ensure by Constraint 4.26 that frames may not overlap if a node is the transmitter for both and by Constraint 4.27 if a node is a receiver of distinct frames.

$$u_{f_i,n} + v_{f_{i'},n} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 \quad \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall n \tag{4.25}$$

$$u_{f_i,T} + u_{f_{i'},T} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 \quad \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall T \tag{4.26}$$

$$v_{f_i,R} + v_{f_{i'},R} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 \quad \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall R \tag{4.27}$$

In order to allow relaying, it should be specified whether a node already received a message, for which we introduce the helper variable $l_{f_i,f_{i'},m_j,n}$. This variable equals 1 if the frame f_i containing message m_j is received by node n at a time before frame $f_{i'}$ starts. To enforce this, we formulate Constraint 4.28:

$$3 \cdot l_{f_i,f_{i'},m_j,n} \leq \phi^<_{f_i,f_{i'}} + s_{f_i,m_j} + v_{f_i,n} \quad \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall m_j, \forall n \tag{4.28}$$

Constraint 4.29 ensures that a message can only be included in a frame if the transmitter of that frame is the origin node of that message, or it received the message in a frame that ended before the start of the current frame.

$$s_{f_i,m_j} + u_{f_i,n} \leq o_{m_j,n} + \sum_{f_{i'} \in \mathcal{F} \setminus f_i} l_{f_i,f_{i'},m_j,n} + 1 \quad \forall f_i, \forall m_j, \forall n \tag{4.29}$$

A successful transmission reaches a receiver before the timeout of the message that it contains, for which we introduce the helper variable $s^<_{f_i,m_j}$. Constraint 4.30 defines that if this variable equals 1, the message m_j is included in the frame. Furthermore, the frame f_i should end before the timeout of m_j, which is ensured by Constraint 4.31.

$$s^{<}_{f_i,m_j} \leq s_{f_i,m_j} \quad \forall f_i, \forall m_j \tag{4.30}$$

$$s^{<}_{f_i,m_j} = 1 \implies t^{start}_{f_i} + \delta^{f}_{f_i} \leq t^{gen}_{m_j} + O \quad \forall f_i, \forall m_j \tag{4.31}$$

Finally, for a receiver R to be successfully reached with message m_j in a frame f_i (represented by $\rho^{s}_{f_i,m_j,R} = 1$), the following three constraints must hold: the receiver should be an intended receiver for the specific message, the receiver should be addressed in the frame and the frame including this message is received before the timeout. This is ensured by Constraint 4.32. Moreover, once the receiver successfully received the message in a frame, delivery of the same message in another frame does not count as a successful transmission anymore, as Constraint 4.33 specifies.

$$3 \cdot \rho^{s}_{f_i,m_j,R} \leq \rho_{m_j,R} + v_{f_i,R} + s^{<}_{f_i,m_j} \quad \forall f_i, \forall m_j, \forall R \tag{4.32}$$

$$\rho^{s}_{f_i,m_j,R} \leq 1 - \sum_{f_{i'} \in \mathcal{F} \setminus f_i} \rho^{s}_{f_{i'},m_j,R} \quad \forall f_i, \forall m_j, \forall R \tag{4.33}$$

All constraints are now formulated. The objective is to maximize the number of intended receivers that receive a mmWave message before the timeout, summed over all messages and all frames, as given in 4.34:

$$\text{maximize} \quad \sum_{f_i \in \mathcal{F}} \sum_{m_j \in \mathcal{M}} \sum_{R \in \mathcal{N}} \rho^{s}_{f_i,m_j,R}. \tag{4.34}$$

This objective might lead to several solutions if the timeout is set loose. In order to optimize it even further, we will introduce a second objective with lower priority. This objective minimizes the delay between message generation and the time at which the last frame ends, for each message. This is formulated as:

$$\text{minimize} \quad \sum_{m_j \in \mathcal{M}} \max_{\forall f_i \in \mathcal{F}} \{ s_{f_i,m_j} \cdot (t^{start}_{f_i} + \delta^{f}_{f_i} - t^{gen}_{m_j}) \}. \tag{4.35}$$

Several other relevant objectives can be thought of, for example to maximize the number of messages that reach all intended receivers. However, these are left out of consideration for this research.
Summarizing, the complete program (omitting the domain constraints) is formulated as on the following page. The MILP presented here is implemented using the mathematical optimization solver Gurobi [30]. Constraints that include a condition can be implemented using indicator constraints. For larger scenarios, the number of input variables and therefore also the computation time of the solver increases rapidly. Some settings are used to speed up the computation time, for example to use more aggressive cuts, but these might lead to numerical issues. Therefore, solutions that include invalid results will be filtered out.

To conclude, a MILP is set up which can be used to find an optimal solution for the scheduling of mmWave transmissions to maximize the number of intended receivers reached given a certain scenario by mathematical optimization. Due to the high complexity of this optimal solution, we will design a lower-complexity heuristic for the problem. The solver will be used later in order to compare the performance of the heuristic algorithm with the optimal solution. The next chapter introduces the devised heuristic for the formulated problem.

$$
\begin{aligned}
&1) \quad \text{maximize} && \sum_{f_i \in \mathcal{F}} \sum_{m_j \in \mathcal{M}} \sum_{R \in \mathcal{N}} \rho^s_{f_i,m_j,R} && \\
&2) \quad \text{minimize} && \sum_{m_j \in \mathcal{M}} \max_{\forall f_i \in \mathcal{F}} \{ s_{f_i,m_j} \cdot (t^{start}_{f_i} + \delta^f_{f_i} - t^{gen}_{m_j}) \} && \\
&\quad \text{subject to} && \sum_{m_j \in \mathcal{M}} s_{f_i,m_j} = q_{f_i} && \forall f_i \\
& && \sum_{T \in \mathcal{N}} u_{f_i,T} = q_{f_i} && \forall f_i \\
& && \sum_{\mathcal{R}_l \in \overline{\mathcal{R}}} w_{f_i,\mathcal{R}_l} = q_{f_i} && \forall f_i \\
& && 2 \cdot w_{f_i,\mathcal{R}_l} \leq u_{f_i,T} + y_{\mathcal{R}_l,T} && \forall f_i, \forall \mathcal{R}_l, \forall T \\
& && 2 \cdot v_{f_i,R} \leq w_{f_i,\mathcal{R}_l} + x_{\mathcal{R}_l,R} && \forall f_i, \forall \mathcal{R}_l, \forall R \\
& && \delta^f_{f_i} = \sum_{\mathcal{R}_l \in \overline{\mathcal{R}}} w_{f_i,\mathcal{R}_l} \cdot \delta^s_{\mathcal{R}_l} && \forall f_i \\
& && t^{start}_{f_i} \geq s_{f_i,m_j} \cdot t^{gen}_{m_j} && \forall f_i, \forall m_j \\
& && \phi^<_{f_i,f_{i'}} = 1 \iff t^{start}_{f_i} + \delta^f_{f_i} \leq t^{start}_{f_{i'}} && \forall f_i, \forall f_{i'}, f_i \neq f_{i'} \\
& && u_{f_i,n} + v_{f_{i'},n} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 && \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall n \\
& && u_{f_i,T} + u_{f_{i'},T} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 && \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall T \\
& && v_{f_i,R} + v_{f_{i'},R} \leq \phi^<_{f_i,f_{i'}} + \phi^<_{f_{i'},f_i} + 1 && \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall R \\
& && 3 \cdot l_{f_i,f_{i'},m_j,n} \leq \phi^<_{f_i,f_{i'}} + s_{f_i,m_j} + v_{f_i,n} && \forall f_i, \forall f_{i'}, f_i \neq f_{i'}, \forall m_j, \forall n \\
& && s_{f_i,m_j} + u_{f_i,n} \leq o_{m_j,n} + \sum_{f_{i'} \in \mathcal{F} \setminus f_i} l_{f_i,f_{i'},m_j,n} + 1 && \forall f_i, \forall m_j, \forall n \\
& && s^<_{f_i,m_j} \leq s_{f_i,m_j} && \forall f_i, \forall m_j \\
& && s^<_{f_i,m_j} = 1 \implies t^{start}_{f_i} + \delta^f_{f_i} \leq t^{gen}_{m_j} + O && \forall f_i, \forall m_j \\
& && 3 \cdot \rho^s_{f_i,m_j,R} \leq \rho_{m_j,R} + v_{f_i,R} + s^<_{f_i,m_j} && \forall f_i, \forall m_j, \forall R \\
& && \rho^s_{f_i,m_j,R} \leq 1 - \sum_{f_{i'} \in \mathcal{F} \setminus f_i} \rho^s_{f_{i'},m_j,R} && \forall f_i, \forall m_j, \forall R
\end{aligned}
$$

Chapter 5

Breakdown of the routing and scheduling problem

In the following chapters, a heuristic algorithm with lower computational complexity will be presented. Before doing so, we first break the problem of routing and scheduling mmWave transmissions down into several steps in this chapter. These steps are called *link assessment*, *graph reduction* and *transmission tree generation*, which will be presented below.

Link assessment

We need to evaluate if a link between a transmitter and a receiver set is possible, meaning that there exists an antenna beam such that the complete set can be reached. Furthermore, if a beam exists, the transmission direction angle and beamwidth should be chosen such that the datarate is maximized. We will call this the *link assessment* process. As explained, the node does this based on position information and antenna and propagation models, rather than beam training via DMG frames. As mentioned before, it has been shown that this method can outperform traditional beamforming in [4].

The complexity of the link assessment problem depends on the number of supported beamwidths, data rates and transmission direction angles. The antenna gain model presented in Equation 3.2 is valid for half-power beamwidths of 15° to 60°. As follows from Table 3.1, we will consider five different beamwidths, such that $\mathcal{W} = \{15°, 30°, 45°, 60°, 360°\}$. For the omnidirectional beam, a transmitting gain of 0 dB is assumed.

Furthermore, only the MCSs of OFDM will be used, leading to 12 different data rates. This already leads to 60 beams with different coverage, which are shown in Figure 5.1. The coverage of beams using the same width are shown with the same color. The beam with largest vertical coverage is achieved with the lowest data rate (693 Mbps), and for each increasing data rate, the vertical coverage decreases. The receiver antenna gain (using beamwidth w_{min}) is already taken into account here.

It can be seen that a wider beamwidth results into more horizontal coverage (meaning perpendicular to the direction of transmission) at the expense of vertical coverage (in the direction of transmission). The needed beamwidth depends on the maximum separation angle the receivers have as seen from the transmitter. However, sorting solely on the maximum width

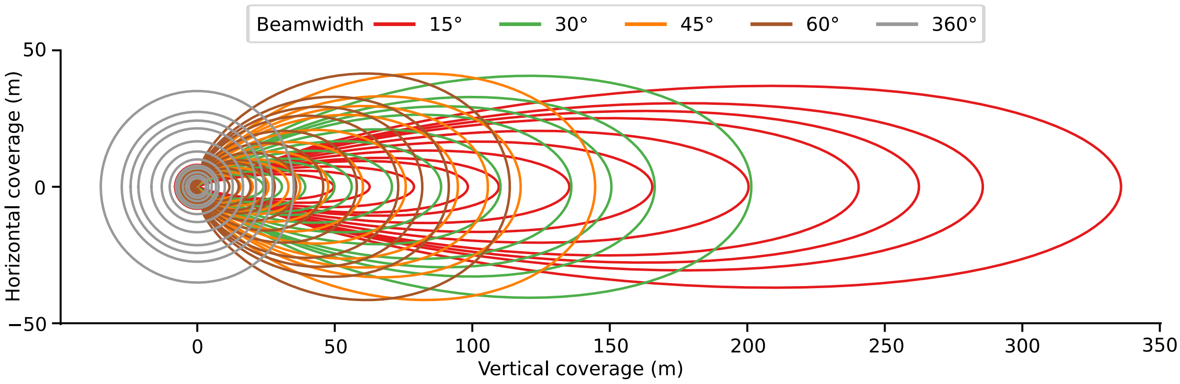


Figure 5.1: Coverage of all supported beams transmitted from (0,0) towards positive x-axis.

of a beam is not applicable here, since it is important at which vertical distance a certain width is achieved. On the other hand, a lower data rate results in increased coverage in all directions. Therefore, if a wider beam is needed for a certain set, lowering the data rate is an option as well.

Besides, the beams can be pointed 360° around. Finding the right transmission angle is also not trivial due to the shape of the beam coverage, except for a beamwidth of 360°. This will be covered in Chapter 6. In the worst case, all possible transmission angles need to be evaluated. We will use a granularity of 1° in the remainder of this thesis, which results into 360 possible angles. In total, in the worst case per link $(|\mathcal{W}|-1)\cdot|\mathcal{D}|\cdot|\mathcal{A}|+|\mathcal{D}|$ combinations need to be evaluated, where we only consider the different data rates for the omnidirectional beam. This will be a total of 17292 combinations considering the aforementioned sets.

To show the complexity of the *link assessment* process, we will give an example. In the scenario of Figure 5.2, two receivers are at the same distance d to a transmitter, at a certain separation angle β as seen from the transmitter. In Figure 5.3, it can be seen which beamwidth is needed to reach both receivers at the same time and achieve the highest data rate. It is shown for a separation angle of 0-179° (in steps of 1°) and a distance of 0-350 m (in steps of 1 m). The maximum distance that can be reached for a certain data rate is shown.

Firstly, it is expected that allowing even more beamwidths will not significantly contribute to better performance, as the figure shows a smooth transition between the beamwidths for the maximum distance covered along a wider separation angle.

Already in this scenario, it is not trivial to determine which beamwidth to use and what the maximum data rate is that can be used. Let alone that there might be multiple receivers that can be at various distances to the transmitter. When the optimal solution needs to be found this is determined exhaustively, meaning that every possible beamwidth, data rate and transmission direction angle is evaluated. However, for a high number of nodes, this will lead to relatively high computation time, so a sophisticated heuristic approach is needed. This will be presented in Chapter 6.

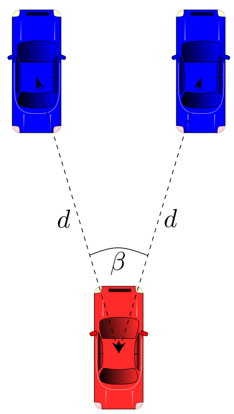


Figure 5.2: Scenario with one transmitter and two receivers at the same distance d to the transmitter, at a certain separation angle β.

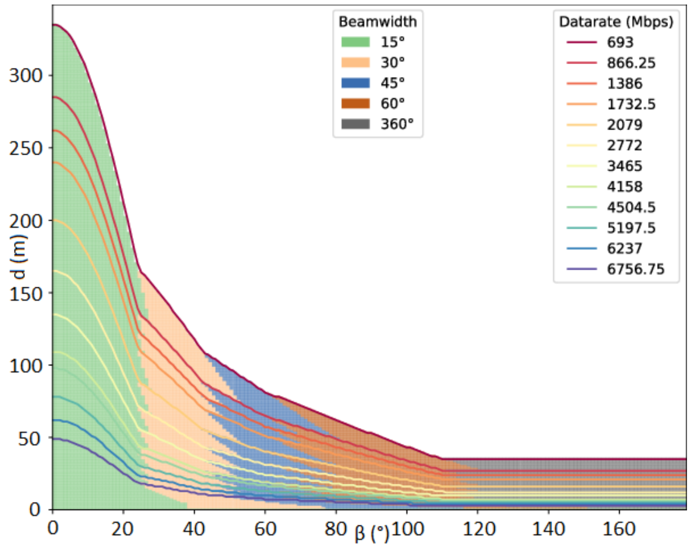


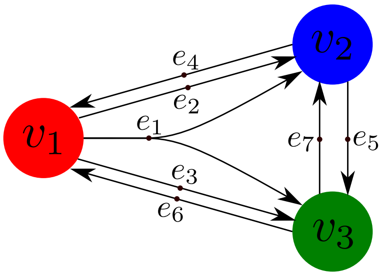
Figure 5.3: Beamwidth that results in the highest data rate at varying separation angle β and distance d to the transmitter, along with maximum distance reached for all data rates.

Graph reduction

Using the link assessment step, it can be determined whether there exists a link between a node and a receiver set. Then, mmWave transmissions using (some of) the possible links should be scheduled. Using sub-6GHz information, there are advantages above geographic routing in which only the location of the destination is known. Namely, a topology of all nodes within sub-6GHz range can be set up. Another approach for determining to determine which transmissions are needed is to consider the choice for the first transmission (hop) only. In this way, a path to the destination is found greedily by forwarding via hops, as is done by Geographic Source Routing. However, the best choice for the first hop is not necessarily the best choice for the complete path. And even worse, it causes the risk of ending up at a point where no further improvement is possible, if at that point there exists no link towards the destination anymore. Hence, since a spatial aware and farsighted vision (namely the sub-6GHz range) is available, a more promising approach is to create a route up to the destinations. Out of the numerous valid routes that exist, we want to find the paths that eventually maximize the number of intended receivers reached before the timeout. Such routing problems are usually represented using graphs. This specific problem can be represented by a *directed hypergraph*, see Figure 5.4.

In a hypergraph, an edge (called a *hyperedge*) can join any number of vertices. A directed hyperedge represents which receivers (vertices) a transmitter will address, if this is possible using a specific beamwidth, data rate and angle combination. Thus, the hypergraph consists of hyperedges with one tail vertex and (possibly) multiple head vertices, which are also called *F-hyperedges*.

In the problem defined before, the origin node needs to reach the intended receivers only, but can use a path via other nodes by relaying. Thus, the problem is to find an optimal interconnect from a transmitter (root vertex) to a given set of vertices (terminals) and a certain



Edge	Cost
e_1	2.5
e_2	2
e_3	2
e_4	2
e_5	1
e_6	2
e_7	1

Figure 5.4: Directed hypergraph of three vertices (represented by $v_1 - v_3$) and seven hyperedges (represented by $e_1 - e_7$), with corresponding cost.

objective function. This is generally referred to as the directed Steiner tree problem or the Steiner arborescence problem. It has been shown that the complexity of the directed Steiner tree problem is already NP-hard [31]. In this specific case, it is the Steiner arborescence problem in hypergraphs [32].
A relevant cost for a hyperedge is the duration of a transmission, which is the message size divided by the data rate corresponding to the hyperedge. Considering the hypergraph in Figure 5.4, the minimum Steiner tree for root vertex v_1 and terminals $\{v_2, v_3\}$ is the edge e_1.

Slight changes in real-life scenarios that are represented by these hypergraphs might lead to different minimum Steiner trees. As an example, consider a hypergraph with three vertices, one hyperedge joining two vertices and three single edges. This is the representation of a scenario with one transmitter and two intended receivers, see Figure 5.5. The cost of the edges differ when the receivers are at different vertical distances (d_1 and d_2, respectively), since different data rates can be achieved. The three options that the transmitter has to reach both the receivers are as follows: (i) transmit to them individually via a unicast, (ii) transmit via a unicast to one, which relays it to the other, or (iii) reach both of them at the same time via a multicast. Figure 5.6 shows which of the options has the lowest total transmission duration for d_1 and d_2 varying from 0 to 350 m, with steps of 1 m. A turquoise section corresponds to the situation in which unicast and relaying result in the same total transmission duration. In the green areas, relaying is the best option and when a section is colored red, multicast is favored. The darker the color, the larger the difference with the second-best option.
It can be seen that when the receivers are roughly at the same distance to the transmitter, it is beneficial to use a multicast. At some points, when using a multicast the datarate to be used is rather low or there does not even exist an antenna beam that reaches both receivers at the same time. In these cases, relaying or unicast is the best option. Only when the receivers are at very distinct distances, letting the transmitter send two unicasts performs equally well as compared to relaying. In some regions, the color is very light meaning that there is marginal performance difference between the first and second option. But, it is hard to generalize where these regions are as we do not see a simple monotone pattern here.
When the hypergraph becomes more complex, the number of options to reach all terminals grows rapidly. If there exists a link between each node and every distinct receiver set, the number of links in the hypergraph equals $N \cdot (2^{N-1} - 1)$. This exponential increase with the

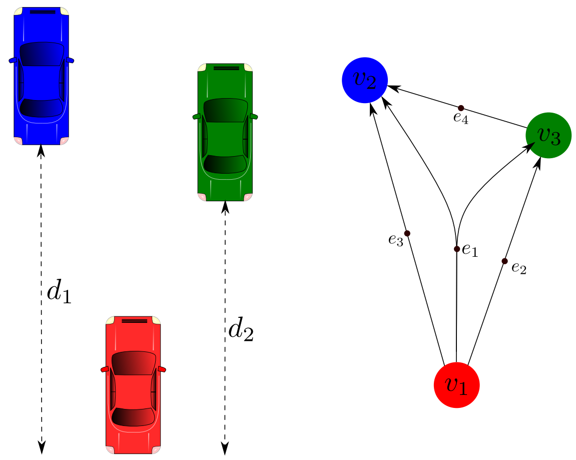


Figure 5.5: Scenario with one transmitter and two receivers at varying vertical distance d_1 and d_2 (left), with its corresponding hypergraph (right).

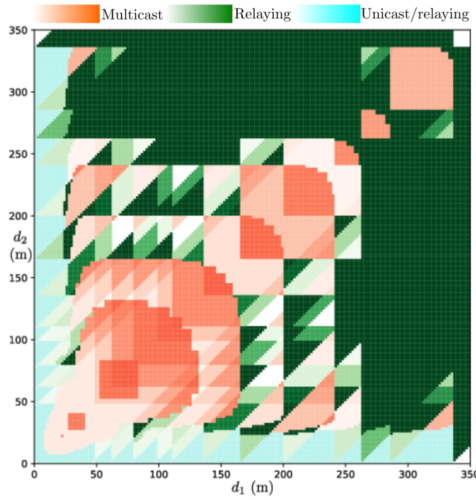


Figure 5.6: Option with lowest total transmission duration for receivers at varying vertical distances.

number of nodes makes it already hard to determine the best option based on some properties of the scenario with a relatively small number of nodes. However, the properties of a scenario can be used to reduce the graph, such that only the vertices and edges that are likely to contribute to the result will be included. For example, nodes located on the opposite side of the intended receivers are unlikely to be in the minimum Steiner tree. However, setting up this graph is not trivial due to the use of relaying and multicast transmissions. We will make use of the path loss from a node to the intended receivers in order to determine whether it should be assessed if a link between a node and receiver set exists. This will lead to a reduced graph with fewer links, as explained in Chapter 7.

Transmission tree generation

Once a reduced graph is obtained, a sophisticated method is needed to find a minimum Steiner tree, for which several heuristics exist. A different problem arises when we want to maximize the number of intended receivers reached before the timeout, instead of minimizing the overall transmission duration. When taking a delay bound into account, finding a minimum Steiner tree is also called the constrained Steiner tree problem [33]. Moreover, since we are dealing with directional antennas, multiple outgoing edges from the same node should be considered subsequently, as a node can only transmit or receive one message at a time. In this case the delay depends on the order at which edges are taken and thus scheduling comes into play, which has several consequences.

For example, using a multicast, the total transmission duration reaching a set of receivers is likely to be lower compared to sequential unicast transmissions, but the delay that an individual receiver experiences might be higher. If later on this specific receiver should relay the message, it is scheduled later than in the unicast case, thereby increasing the overall delay, which might result in fewer receivers reached before the timeout. For example, consider the

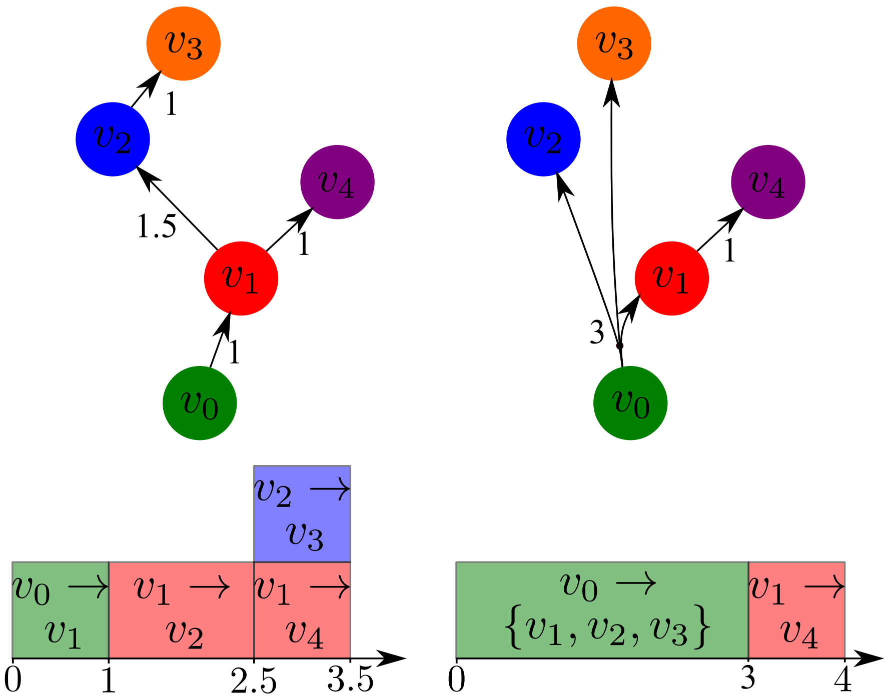


Figure 5.7: Unicast (left) and multicast tree (right) and their corresponding schedules.

trees and their schedules in Figure 5.7. The origin node is represented by v_0, which needs to reach the receivers v_1, v_2, v_3 and v_4. The durations of the transmissions represented by the edges are displayed next to each edge.

On the left, only unicast transmissions are used, whereas on the right, a multicast transmission is used to cover vertices v_1, v_2 and v_3 all at the same time. Although the sum of transmission durations of the multicast tree is lower than that of the unicast one (4 instead of 4.5), the unicast transmissions can be scheduled with lower delay, because concurrent transmissions are possible. This also means that all nodes are already idle at 3.5, whereas on the right this is only at 4. On the other hand, if v_2 or v_3 needs to transmit again directly afterwards, the multicast schedule is more favorable. Thus, the scheduling is of great importance in finding a tree. Eventually, a transmission tree is found, specifying the frames that should be used and the order in which they are transmitted, which is explained in Chapter 8. In the remainder of this thesis, we will present different options and methods for the steps described above. Next to that, we will compare heuristic methods with an exhaustive search or optimal solution. Figure 5.8 provides an overview of the considered steps and options.

The aforementioned steps are required to solve the problem for one message. When multiple messages are considered, routes need to be found while the scheduling of transmissions for all messages should be taken into account. Following the analogy of the hypergraph, multiple Steiner trees need to be packed, in which some vertices should be disjoint during certain time intervals. Since it is not known upfront when messages arrive, scheduling should be decided upon dynamically. Moreover, to achieve better joint performance, it will be addressed in a distributed way, where the additional sub-6GHz layer provides means for. The implementation of this will be addressed in Chapter 10.

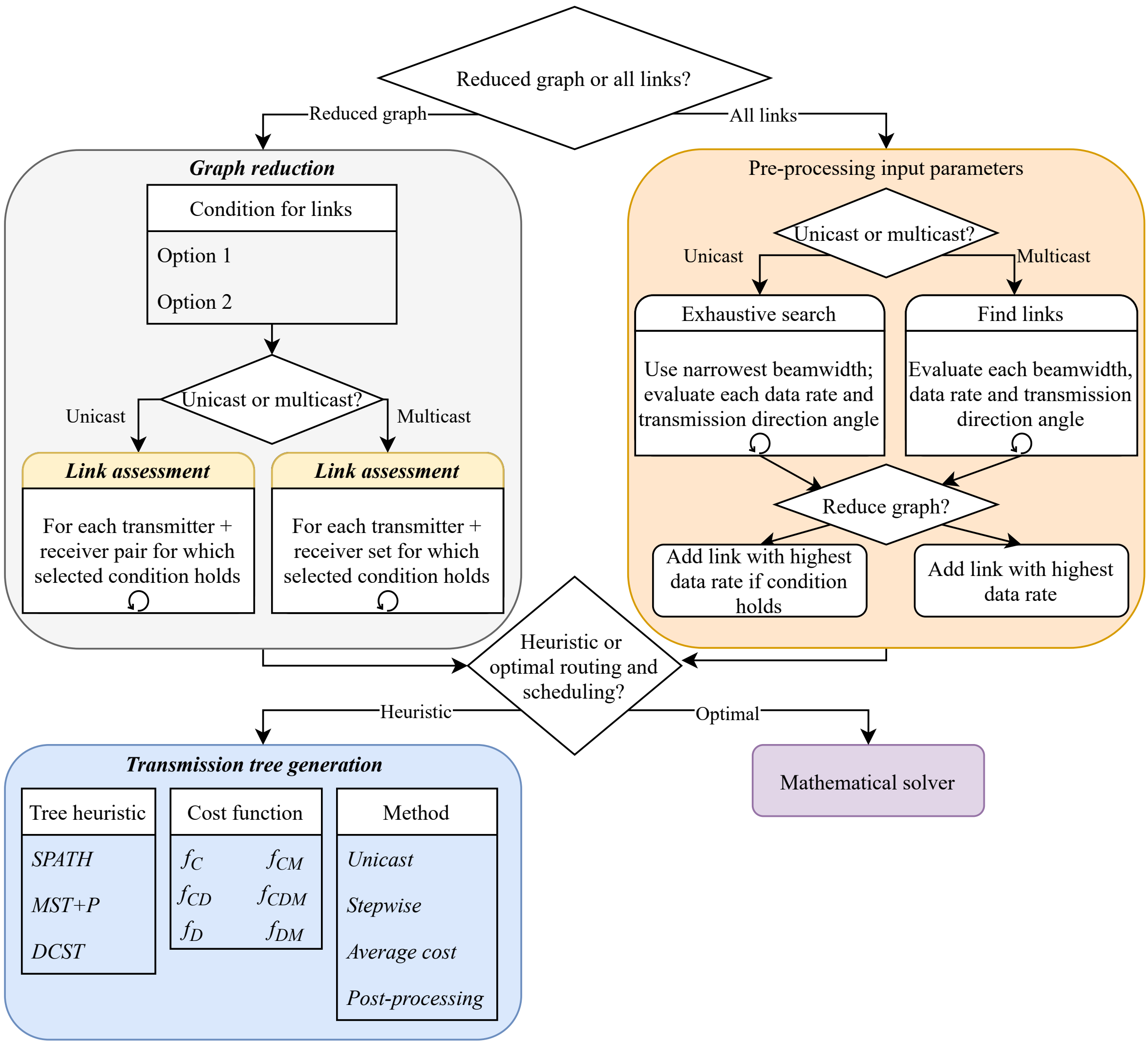


Figure 5.8: Different options and steps for solving a scenario as presented in this thesis.

Chapter 6

Link assessment

Before a node is able to make decisions on transmissions to send, it should determine the links to possible receiver sets, and which beamwidth, data rate and transmission direction angle should be used. In other words, the hyperedges and corresponding cost of the directed hypergraph should be determined. The cost corresponding to an edge will be set equal to the duration of the transmission. This is determined by the highest supported data rate, such that the coverage is still sufficient.

Realistic antenna models with various beamwidths, like the one presented in Chapter 3, and the use of various data rates make determining the link quality based on position information a problem on its own. Given the high complexity as presented in Chapter 5, the best combination of direction angle, beamwidth and data rate, if it exists, will be determined heuristically. Afterwards, we will assess the performance of the heuristic approach as compared to the exhaustive search method.

The link assessment procedure can be separated into two parts, namely in the case of a unicast (when the receiver set consists of one node) and a multicast transmission. The procedure for both parts will be described next.

6.1 Unicast

For a unicast transmission, the transmitter will always use the narrowest beamwidth ($w_{\min}$), so that the data rate is maximized. Furthermore, the transmission direction angle will be the one closest to the angle between transmitter and receiver, as calculated from their positions. With a granularity of 1° for the transmission direction, the observed angle of the transmitting beam was rounded to 0°, resulting in a gain of $G_0(w_{\min})$, like for the receiving antenna beam. The data rate to be used will be the maximum available given that the estimated received power is higher than or equal to the receiver sensitivity. The path loss between transmitter T and receiver R is needed for this, which is calculated using the distance and the amount of obstacles between them. The data rate r^* to be used is then given by:

$$r^* = \max_{r \in \mathcal{D}} \{ r \mid P_T + G_0(w_{\min}) + G_0(w_{\min}) - PL_{T \to R}(d_{T \to R}) \geq P_{sens}(r) \}. \tag{6.1}$$

The value for r^* will be the output of a one-dimensional sorted look-up table listing the minimum received power needed for all data rates as its entries. This lookup table is referred to as `lookUpDatarate` further on.

6.2 Multicast

In case of a multicast transmission, the selection of the beamwidth and data rate is based on three properties of the receiver set. These properties are chosen such that they represent the topology of the receiver set best, while minimizing the computational costs. Using these properties, we will create a three-dimensional lookup table, that gives the beamwidth and data rate to use for the receiver set of interested, which will be called `lookUpBeamwidthDatarate`.
The first property that is used is the maximum separation angle ($maxSepAngle$) of all pairs of receivers as seen from the transmitter. For example, consider the situation of Figure 6.1, which illustrates the coverage of the optimal beam from transmitter 0 to receivers 1, 2, 3 and 4. The pair with maximum separation angle as seen from the transmitter consists of receiver 1 and 4.

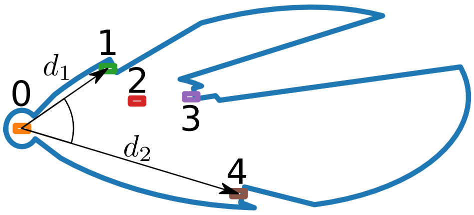


Figure 6.1: Situation with transmitter 0 and receivers 1, 2, 3 and 4, along with the coverage of the optimal beam for 0 to reach all receivers.

The second property incorporates the distances between the transmitter and each of the receivers of the aforementioned pair, displayed as d_1 and d_2 in Figure 6.1. The coverage perpendicular to the direction of transmission slowly increases for a larger vertical distance from the transmitter, after which it decreases again. For each beamwidth and data rate, this gradient is different, as can be seen in Figure 5.1. When incorporating both distances, the varying horizontal coverage is taken into account on both sides of the beam. Due to the symmetry of the antenna beam, it is irrelevant whether the receiver with largest distance to the transmitter is on the right and the one with smallest distance on the left, or vice versa. Therefore, only the unique combinations of distances to the pair need to be evaluated.
The last property is the maximum path loss ($maxPathloss$) from the transmitter to any of the receivers in the set. This property is included as the pair of maximum separation does not necessarily contain the receiver with maximum path loss, whereas this receiver might cause that the beam does not suffice, as its received power is too small. For example, in the case of Figure 6.1, the receiver with the minimum received power is node 3, due to the obstacle formed by 2. Moreover, the receivers in the pair of maximum separation might be blocked by obstacles themselves, causing that the beam is deformed and incorporating the distance to the transmitter is not sufficient for determining whether the beam covers both receivers.

The lookup table is generated by creating multiple topologies with four nodes; one transmitter and three receivers. The receivers are placed in such a way that all possible values (with a certain granularity) for the aforementioned properties are evaluated. Two nodes, representing the pair of maximum separation, are placed at varying separation angles and distances from the transmitter. The third will be placed in the direction of transmission, at a distance such that it experiences the maximum path loss. At each topology that is set up, the optimal beam is determined by an exhaustive search. The beamwidth and data rate of this beam are inserted at the location in the lookup table corresponding to the properties of this topology. Then, during the link assessment step, for a receiver set of interest, the three properties are calculated and rounded towards the granularity used and these values are looked up in the table. From this follows that the smaller the granularity, the better the performance of the *link assessment* algorithm. However, it influences the size of the lookup table and the computation time for creating it. The value of the properties vary from the minimum and maximum value they can take such that an antenna beam exists. In Table 6.1, the minimum, maximum and step size for the parameters to calculate the entries for the tables `lookUpDatarate` (for which only the path loss is needed) and `lookUpBeamwidthDatarate` that are used throughout this thesis are given. The maximum separation angle is 180°, when the receivers can be covered by the side lobe or the omnidirectional beam. The maximum distance is calculated from the maximum path loss that results in sufficient received power (as calculated by Equation 3.3) for the lowest data rate considering zero obstacles, using the narrowest beamwidth $w_{\min}$ and observed angle $\alpha = 0°$.

Parameter	**Minimum**	**Maximum**	**Step size**	**Unit**
$maxSepAngle$	0	180	3	°
$distances$	0	340	5	m
$maxPathloss$	0	126	$PL(d)$ with d step of 1 m	dB

Table 6.1: Minimum, maximum and step for parameters to calculate the lookup tables.

From the created lookup table, the beamwidth and data rate to use for a receiver set are derived. Subsequently, the optimal transmission direction angle can be computed. It may seem straightforward to point the beam halfway between the pair of maximum separation. However, this does not guarantee to cover all receivers, because the width varies along the vertical direction, causing that receivers at different vertical distances have different margins to the side of the beam. Therefore, for the calculation of the transmission direction angle, for all receivers, the margin towards the side of the beam coverage is determined, when the beam is pointed directly towards this receiver. See Figure 6.2 for an example when the beam is pointed towards receiver 1. The coverage shown is again calculated including maximum receiver antenna gain.

As a following step, the beam will be rotated by an angle equal to the determined margin, such that one receiver will be just inside the beam. It is rotated in the direction where most of the receivers are, creating the largest coverage for the remaining receivers. The angle resulting from this is calculated by the function `calcAngleFromMargin(`T, R, w, r`)`. In Figure

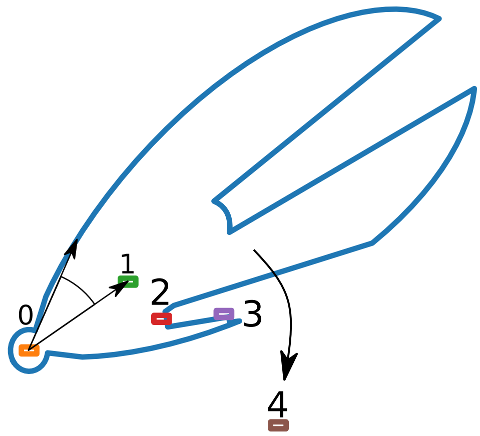


Figure 6.2: Coverage of the beam pointed towards 1 and the margin towards the side of the coverage.

6.2, the beam will be rotated towards 4, such that both 1 and 4 are just inside the beam coverage.
If the values of w and r resulting from the lookup table form a beam with which all receivers can be reached, a transmission direction angle will be found that covers them in this way. However, even if the properties have valid values such that they can be looked up in the table, it is not guaranteed that all receivers can be reached with the resulting parameters. Firstly, this can be caused by rounding of the properties, such that the resulting parameters of the lookup table are not valid. Furthermore, it can be caused by a receiver that is not in the pair of maximum separation, but it is located at a distance where the beam is narrow, causing it to fall out of coverage. Lastly, if the receiver with maximum path loss observes the transmitting beam at a rather large angle after the choice for a, the antenna gain might not suffice for it to receive the signal, because the value in the lookup table is calculated for a receiver in the direction of transmission. These imperfections lead to performance loss, caused by the choice for including only these three properties, which cannot cover all possible topologies. On the other hand, once the parameters a, w and r are calculated, an additional check can be performed at a relatively low cost in order to filter out falsely assigned beams. Only if for all receivers in the receiver set their received power is at least the required sensitivity the selected combination is used. This is evaluated by the function `checkAllReachable(`T, $\mathcal{R}$, w, r, a`)`. If for none of the receivers rotating the beam with a margin such that it is just within the beam leads into coverage for all receivers, the conclusion is that no beam is possible.

The procedure for determining the transmission angle, beamwidth and data rate to use for a transmitter and a certain receiver set is summarized in Algorithm 1.
As the maximum separation between all pairs of receivers needs to be determined, the complexity of this algorithm is equal to $\mathcal{O}(|\mathcal{R}|\cdot(|\mathcal{R}|-1))$. Thus, instead of the number of available beamwidths, data rates and angles, the complexity of assigning a link depends on the size of the receiver set for the heuristic method. Typically, multicasts will need to be evaluated for a limited receiver set size, which makes the heuristic favorable in terms of complexity.

Algorithm 1: Determine a, w and r for a transmitter T and receiver set $\mathcal{R}$.

Input : Transmitter T, receiver set $\mathcal{R}$ and coordinates of all discovered nodes.
Output: Transmission angle (a), beamwidth (w) and data rate (r) to use.

if $|\mathcal{R}| \geq 2$ **then** `/* Multicast */`
- $maxSepAngle \leftarrow$ Maximum separation angle between all pairs in $\mathcal{R}$.
- $d_1, d_2 \leftarrow$ Distances to pair of maximum separation.
- Determine the obstacles between T and each receiver.
- $maxPathloss \leftarrow$ Highest path loss for all receivers, taking obstacles into account.
- Round *maxSepAngle*, *distances* and *maxPathloss* towards granularity used in lookup table.
- w, $r \leftarrow$ `lookUpBeamwidthDatarate`(*maxSepAngle, distances, maxPathloss*)
- **foreach** $R \in \mathcal{R}$ **do**
  - $a \leftarrow$ `calcAngleFromMargin`(T, R, w, r)
  - $result \leftarrow$ `checkAllReachable`(T, $\mathcal{R}$, w, r, a)
  - **if** $result ==$ **True then**
    - **return** a, w, r
- **return** 0, 0, 0 `/* No beam possible. */`

else `/* Unicast */`
- $a \leftarrow$ Angle towards receiver, rounded towards transmission angle granularity.
- Determine distance and obstacles between T and the receiver.
- $PL_{T \to R} \leftarrow$ Pathloss from T to receiver, taking obstacles into account.
- $r \leftarrow$ `lookUpDatarate`($PL_{T \to R}$)
- **return** a, $w_{\min}$, r

6.3 Evaluation

In this section, the performance of the heuristic algorithm will be evaluated by checking whether the beam parameters are correctly assigned for multiple artificial topologies. Furthermore, its computation time will be evaluated against the exhaustive search. Then, we will introduce realistic topologies that will be used in the remainder of this thesis. We will use these to show the performance degradation when the input for the mathematical optimizer is generated via the heuristic method.

Performance of beam parameters assignment in artificial scenarios

The performance of the multicast link assessment part of Algorithm 1 is evaluated by creating artificial scenarios with one transmitter and two to four receivers. The first receiver is placed randomly around the transmitter, while assuring it is in range of the beam with minimum data rate and beamwidth. The following receivers are placed closely around the other receivers such that it is likely that they are reachable using one transmission. The horizontal position may not differ more than 42 m from the already placed receivers. The

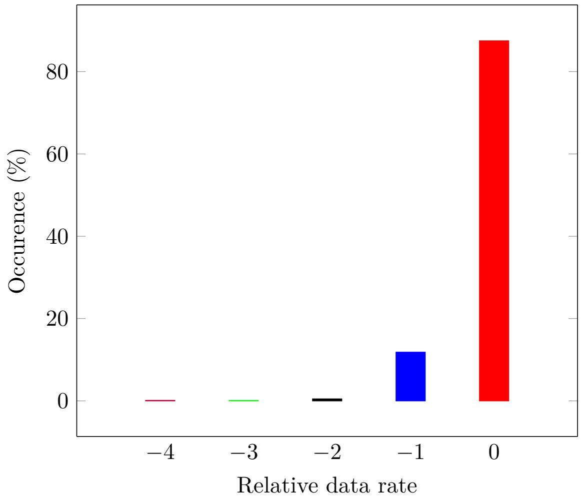


Figure 6.3: Distribution of the relative data rate selection for true positives.

maximum vertical position is such that the receiver can be reached by the transmitter using the beam with maximum data rate and minimum beamwidth, if pointed in the direction of the other receivers. For each of the generated scenarios, an optimal beam is looked for by an exhaustive search and a beam will be searched for using `lookUpBeamwidthDatarate`. If both methods find a beam that reaches all receivers, this is defined as a true positive. If using the exhaustive search a beam is found, as well as when using `lookUpBeamwidthDatarate`, but the check `checkAllReachable()` fails, this outcome is considered as a false positive. If none of the methods finds a beam, it is called a true negative. Lastly, if using the exhaustive search a beam is found, but using the look-up table no beam is found, a false negative result is obtained.

Evaluating 10,000 of these scenarios results in a false positive rate of 2.45% and a false negative rate of 15.48%. However, when the additional check `checkAllReachable()` is used in the algorithm, all false positives will become true negatives. Thus, in 15.48% of the cases no beam is found by the heuristic method, while there does exist one that reaches all receivers. Still, when a beam is found, the heuristic method may assign a data rate that is lower than the one assigned via an exhaustive search. The distribution of relative data rate choices in case of a true positive outcome is shown in Figure 6.3. A value of -1 means that the heuristic method chose a data rate that is one lower (in the set $\mathcal{D}$) than the one assigned by the exhaustive search.

It can be seen that in around 87% of the cases, the same data rate as the exhaustive search is selected. Furthermore, in around 12% of the cases, one data rate lower than the optimal one is chosen. In less than 1% of the cases a data rate is chosen that is even lower than this. So, the overall performance degradation is that about 15% of the beams are not detected and 13% of the detected beams use a data rate that is too low.

Next, the exhaustive search and heuristic methods are evaluated in terms of their computation times. Both methods are implemented using Python and evaluated using 5,000 of the aforementioned artificial scenarios on a laptop with an Intel i7-6700HQ processor with 16 GB

RAM. The average computation time per scenario of five runs is taken. This results into 55.6 ms for the exhaustive search, whereas using the lookup table this is 0.842 ms. Thus, for receiver sets of two to four receivers, finding a beam using the heuristic method takes on average only 1.5% of the time required by an exhaustive search.

Performance degradation in realistic scenarios

For performance evaluation, we will make use of realistic topologies from a highway scenario several times in the remainder of this thesis. These topologies are derived from snapshots of the five-lane US Highway 101, captured in June 2005, as presented in [34]. This dataset includes, amongst others, the positions of vehicles at every 100 ms during three 15 min intervals. In total the considered segment is 640 m long, but only the first part (starting North) is considered, such that a fixed number of vehicles is included in a scenario. For small scenarios, the vehicles will quickly leave the considered section and others will join, resulting in snapshots with distinct vehicles after a short time interval. However, due to the way how traffic behaves, the positions might be related to each other. Therefore, the snapshots with 100 ms separation cannot be considered completely independent. However, for scenarios with a limited number of nodes, we can evaluate a large number of snapshots. For larger scenarios, we will take snapshots with at least 20 s separation, but due to the limited dataset size we will evaluate 100 different snapshots.

For the following evaluation, in total 1000 snapshots at 100 ms intervals containing five nodes were considered. For each snapshot, a beam was searched via the exhaustive search and the heuristic method for all possible distinct receiver sets for each transmitter. The number of distinct receiver sets per transmitter is equal to the size of the powerset of the remaining nodes (except the empty set), which is 15 in this case. Compared to the exhaustive approach, the heuristic method produced 34.68% fewer links. This is significantly more than the percentage of false negative beams derived before, which is caused by the fact that multiple distinct receiver sets may use the same calculated properties. For example, if for a receiver set $\{1, 2\}$ no beam could be found, there will also be no beam for the receiver set $\{1, 2, 3\}$ if node 3 is not in the pair of maximum separation or the receiver with maximum path loss. However, since in the final mmWave schedule only non-overlapping receiver sets will be included, it is expected that the percentage of links that are selected by the optimal solution, which are not detected by the heuristic link assessment process, is lower. To confirm this, from each snapshot a scenario is created, in which the northernmost vehicle is designated as transmitter and from the four remaining nodes, three are randomly assigned as intended receiver. Then, an optimal solution for scheduling one mmWave message with a timeout of 150 ms is found by solving the MILP using Gurobi. The solver uses the links determined by an exhaustive search, as explained in Section 4.1. Due to numerical issues, for only 900 scenarios the optimal solution was obtained. From the links chosen by the optimal solution, on average 15.99% were not detected by the heuristic link assessment process, which is indeed lower than the total percentage of links that are not included. Nevertheless, it is a significant performance degradation already at the first step, as a result of using a heuristic method. The influence of not detecting these links on the overall performance of the heuristic algorithm will be evaluated later.

Concluding, we designed a link assessment procedure using lookup tables. For a unicast link, only the path loss is needed to determine which datarate should be used. For a multicast, the beam parameters are based on three properties of the receiver set to cover all possible topologies best. These properties are the maximum separation angle between all pairs of receivers as seen from the transmitter, the distances to each of this pair and the maximum path loss to any of the receivers. The procedure has a lower complexity and uses only 1.5% of the computation time needed by the exhaustive search method for receiver sets of two to four receivers. This comes, however, at the cost of performance degradation, namely about 16% of the links used by the optimal solution are not detected by the heuristic procedure.

Chapter 7

Graph reduction

Using Algorithm 1, a node can create an overview of the network topology, which includes all possible unicast and multicast links with their corresponding costs. Given the exponential increase of the number of links with the number of considered nodes, using such a topology might lead to a high computation time for creating and evaluating it. Already at this stage, the number of calculations can be reduced if specific nodes or links that will likely not contribute to the result are excluded. In this way, a relevant graph $\mathcal{G}$, with useful nodes and edges $\mathcal{E}$ can be created, where an edge $E(T \to \mathcal{R})$ represents a link between a transmitter T and a receiver set $\mathcal{R}$, which has a cost $E_C(T \to \mathcal{R})$ that is equal to the transmission duration. To determine which links should be included in the reduced graph and which not, we will make use of the intended receivers, as we will explain first. Afterwards, we will evaluate the performance of the graph reduction step by determining the reduction in number of links, and the percentage of links used by the optimal solution that are also included in the resulting graph.

7.1 Procedure

As a first step in reducing the graph, only the nodes that have a lower path loss to any intended receiver than the origin node will be used. This is done as it is unlikely that a better path exists via a node that has a higher path loss, since the total duration of transmissions needed to reach the intended receiver will likely be higher. The nodes that have a lower path loss to any intended receiver than the origin node are called Multipoint Relays (MPRs), like in Optimized Link State Routing Protocol (OLSR) [35]. Furthermore, the origin node and all intended receivers are MPRs themselves. Only nodes that are an MPR will be included in the graph.
In addition, we introduce the notion of Multipoint Relay Destinations (MPRDs). The MPRDs of a node are the intended receivers which can be reached by using this node as MPR. Furthermore, intended receivers are MPRDs for themselves and the origin node has all intended receivers as its MPRDs.
With the MPRs, we identified the nodes that will be in the graph. Now, let us introduce how we identify the links that will be in the reduced graph. Only if certain conditions apply,

a link between an MPRs and a (set of) MPR(s) is evaluated by applying Algorithm 1, the link assessment step. In this way, fewer receiver sets will need to be evaluated and also fewer links will appear in the hypergraph. The conditions for evaluating a link are different for a unicast and a multicast link, which will be explained next.

Unicast condition

A unicast link between two MPRs, specifically from MPR_T to MPR_R, is evaluated only if they have an MPRD in common. As the origin node has all intended receivers as MPRDs and the intended receivers are an MPRD for themselves, links between them will be allowed. Furthermore, relaying nodes might form a path with lower cost towards an intended receiver in this way. However, an additional condition is that the path loss between MPR_R and at least one of the shared MPRDs should be lower than or equal to the path loss between MPR_T and this specific MPRD. If this does not hold, MPR_R will never have a lower transmission duration to one of the MPRDs of MPR_T, and thus this link can be neglected. Lastly, the path loss from MPR_T to MPR_R should be smaller than or equal to the path loss from MPR_T to the specific shared MPRD. Otherwise, it would be favorable to let MPR_T directly transmit to the specific MPRD. The condition for a link from MPR_T to MPR_R is formally given by:

$$
\begin{aligned}
(\exists MPRD \in MPRDs(MPR_T) \quad | \quad & MPRD \in MPRDs(MPR_R) \quad \wedge \\
& PL_{MPR_R \rightarrow MPRD} \leq PL_{MPR_T \rightarrow MPRD} \quad \wedge \\
& PL_{MPR_T \rightarrow MPR_R} \leq PL_{MPR_T \rightarrow MPRD}). \qquad (7.1)
\end{aligned}
$$

For example, consider the reduced graph in Figure 7.1. In the figure, a node is colored red if it is an intended receiver, otherwise it is colored blue. Furthermore, each node is labeled with its ID followed by the list of its MPRDs.

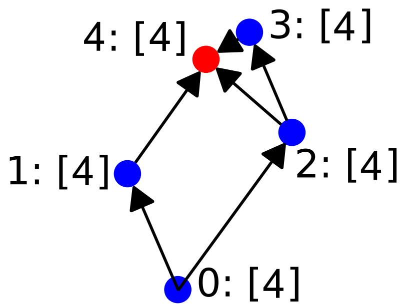


Figure 7.1: Reduced graph with intended receiver 4 and origin node 0.

Node 0 is the origin node, so it has the intended receiver 4 as MPRD. Node 0 cannot reach 3 or 4 directly. Each of the nodes 1, 2 and 3 are an MPR, because their path loss to node 4 is lower than from node 0 to 4. Besides, node 4 is an MPR for itself. There exist links between each of the MPRs and node 4, because in this specific case the MPRD is the receiving MPR itself. Although node 2 has a lower path loss to node 4 than node 1 has, there is no link from 1 to 2, because the path loss of that link is higher than the path loss from 1 directly towards 4. The same holds for the link between 1 and 3. On the other hand, the pathloss between node 2 and 3 is lower than from 2 to 4, therefore the link between 2 and 3 is included in the reduced graph.

Multicast condition

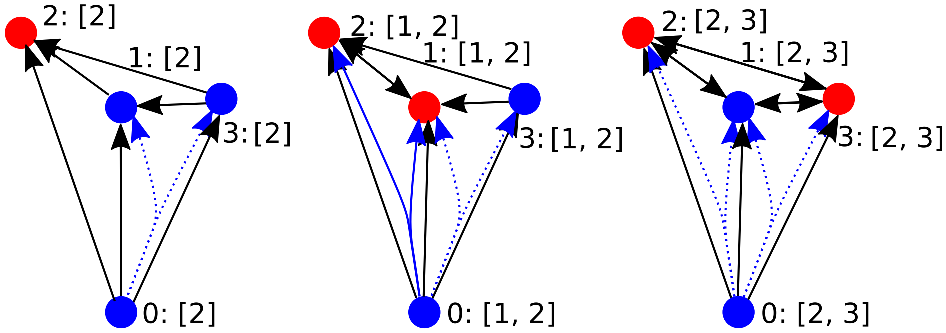
For evaluating a multicast link, we will present two options that pose a trade-off between computational complexity and performance. The first option is given as follows. Consider a multicast link from MPR_T to a set of MPRs, $\mathcal{MPR}_R$. Each MPR in $\mathcal{MPR}_R$ should either be an MPRD for MPR_T itself, or should have an MPRD that is in the MPRD set of MPR_T, and this MPRD is not included as receiver in this link. Furthermore, for this specific MPRD, Condition 7.1 should also hold. This ensures that every receiver of this link is either a destination, or may contribute to reaching one via relaying. The condition of option 1 for a multicast link from MPR_T to $\mathcal{MPR}_R$ is formally given by:

$$\begin{aligned}(MPR_R \in MPRDs(MPR_T) \quad \vee \\ (\exists MPRD \in MPRDs(MPR_R) \mid MPRD \in MPRDs(MPR_T) \wedge MPRD \notin \mathcal{MPR}_R) \\ \forall MPR_R \in \mathcal{MPR}_R).\end{aligned} \tag{7.2}$$

The second option is more strict. Namely, if an MPR in $\mathcal{MPR}_R$ is not in the MPRD set of MPR_T, it should have a **unique** MPRD among all MPRDs of the receivers in this link, that is in the MPRD set of MPR_T. The incentive behind this is that every MPR should have an MPRD that is not handled by the other receivers of the multicast link. The condition of option 2 for a multicast link from MPR_T to $\mathcal{MPR}_R$ is formally given by:

$$\begin{aligned}(MPR_{R_1} \in MPRDs(MPR_T) \quad \vee \\ (\exists MPRD \in MPRDs(MPR_{R_1}) \mid MPRD \in MPRDs(MPR_T) \wedge MPRD \notin MPRDs(MPR_{R_2}), \\ \forall MPR_{R_2} \in \mathcal{MPR}_R, MPR_{R_2} \neq MPR_{R_1}), \forall MPR_{R_1} \in \mathcal{MPR}_R).\end{aligned} \tag{7.3}$$

As an example, consider the reduced graphs with a common topology, but with different sets of intended receivers in Figure 7.2. In the figure, single edges are drawn in black and hyperedges in blue. Hyperedges that will only be included when using option 1 are displayed with dashed arrows.



a) $\mathcal{R}_m = \{2\}$ b) $\mathcal{R}_m = \{1, 2\}$ c) $\mathcal{R}_m = \{2, 3\}$

Figure 7.2: Reduced graphs for different sets of intended receivers.

In a), there is no link from 1 to 3, since the path loss from 1 directly to 2 is less. Furthermore, there is no hyperedge from 0 to {1,2}, because then the MPRD of 1 would already be included in the multicast. In case option 1 is used, there exists a multicast link from 0 to {1,3}, because both nodes have an MPRD that is not included in this multicast (namely node 2).
In b), there is a hyperedge from 0 to {1,2}, because both are in the MPRD set of 0. In case option 1 is used, there exists a multicast link from 0 to {1,3}, for the same reason as in a). Lastly, if it would exist, a multicast link from 3 to {1,2} was also allowed.
In c), node 1 is an MPR for both node 2 and 3, because its path loss to them is lower than that of node 0. Also, for option 1 there exist hyperedges from 0 to {1,2} and from 0 to {1,3}, because node 1 has an MPRD (either 2 or 3), which is not yet included in the link and is in the MPRD set of node 0. However, these would not be included using option 2, because 2 is also an MPRD for 3 and vice versa, thus these are not unique. If it would exist, a multicast link from 0 to {2,3} would be included in the reduced graph for both options.
The consequence of option 2 is that multicast transmissions in which a receiver has a lower path loss to an MPRD, but which is not a unique MPRD, are neglected. This choice is made since usually the difference between path loss to MPRDs within a multicast transmission is limited, because the receivers should be rather close to each other in order to be included in a multicast transmission. On the other hand, in order to schedule concurrent transmissions, there should be multiple relayers available. This is still possible using option 2, but only if these relayers have different MPRDs, whereas using option 1, this is not a restriction. Furthermore, using option 2, still separate links containing only one of the MPRs that serve the same MPRD will be included. Since if a beam exists that covers a receiver set also links that address only a part of this set are valid, this does not lead to excluding links reaching this MPRD.

The procedure of generating the reduced graph for a given set of intended receivers is given in Algorithm 2.
In lines 2-10 the useful nodes of the graph, namely the MPRs, and their MPRDs are determined. Then, for each MPR the powerset of the other MPRs is determined (except the empty set), in order to evaluate all possible distinct receiver sets. If this leads to a set that is disproportionately large, it can be limited by calculating the powerset of only the reachable receivers for each MPR. However, then first it should be determined whether a receiver is reachable for a certain MPR, meaning that the beam of farthest coverage reaches it. For a rather small set of nodes, this results in unnecessary overhead. Only at the point where the latter computational overhead outweighs the overhead induced by using simply the powerset of all MPRs, this should be used.
Subsequently, in lines 11-17 it is evaluated if a link from a MPR to a receiver set should be evaluated based on the aforementioned conditions. If the conditions apply, Algorithm 1 is consulted, resulting in an edge with cost equal to the transmission duration, if the link exists. When all MPRs are assessed, the reduced graph is fully determined.
Since for every MPR, the powerset of the remaining MPRs should be evaluated, the com-

Algorithm 2: Generate reduced graph $\mathcal{G}(\mathcal{MPR}, \mathcal{E})$.

Input : Origin node n_{m_j}, intended receivers $\mathcal{R}_{m_j}$ and coordinates of nodes $\mathcal{N}$.

Output: Reduced graph $\mathcal{G}(\mathcal{MPR}, \mathcal{E})$ and the MPRDs for each MPR.

$\mathcal{E} \longleftarrow \emptyset$ `/* Initially no edges in the reduced graph. */`

$\mathcal{MPR} \longleftarrow \{n_{m_j}\} \cup \mathcal{R}_{m_j}$ `/* Origin node and intended receivers are MPRs. */`

$\mathcal{MPRD}(n_{m_j}) \longleftarrow \mathcal{R}_{m_j}$ `/* Origin node has intended receivers as MPRDs. */`

foreach $n \neq n_{m_j} \in \mathcal{MPR}$ **do**
- $\mathcal{MPRD}(n) \longleftarrow \{n\}$ `/* MPR has itself as MPRD. */`

foreach $R \in \mathcal{R}_m$ **do** `/* Determine MPRs and MPRDs. */`
- **foreach** $n \in \mathcal{N}, n \neq R$ **do**
  - **if** $PL_{n \to R} < PL_{n_{m_j} \to R}$ **then**
    - $\mathcal{MPR} \longleftarrow \mathcal{MPR} \cup \{n\}$
    - $\mathcal{MPRD}(n) \longleftarrow \mathcal{MPRD}(n) \cup \{R\}$

foreach $MPR \in \mathcal{MPR}$ **do** `/* Determine links between MPRs. */`
- $\wp(\mathcal{MPR}) \longleftarrow$ `Powerset`$(\mathcal{MPR} \setminus MPR)$
- **foreach** $\mathcal{R} \in \wp(\mathcal{MPR})$ **do**
  - **if** $|\mathcal{R}| < 2$ **then**
    - $result \longleftarrow$ Condition 7.1 holds.
  - **else**
    - $result \longleftarrow$ Condition 7.2 or 7.3 holds.
  - **if** $result ==$ **True then**
    - $a, w, r \longleftarrow$ `Algorithm 1.`
    - **if** $r > 0$ **then** `/* Link exists. */`
      - $E_C(MPR \to \mathcal{R}) \longleftarrow \frac{S}{r}$ `/* Transmission duration. */`
      - $\mathcal{E} \longleftarrow \mathcal{E} \cup \{E(MPR \to \mathcal{R}), E_C(MPR \to \mathcal{R})\}$

plexity of the algorithm equals $\mathcal{O}(|\mathcal{MPR}| \cdot (2^{|\mathcal{MPR}|-1} - 1))$. If no reduced graph is used, this should be evaluated for all nodes in order to obtain all links. Another benefit of the reduced graph is that if the condition fails, Algorithm 1 does not have to be applied. And more importantly, it results in fewer available links for creating the transmission tree. The influence of this step in realistic scenarios is presented next.

7.2 Evaluation

We will assess the reduction in the number of links (both unicast and multicast) in the reduced graph, either using option 1 or 2, as compared to a graph with all links. The latter is generated following the procedure on the right in Figure 5.8. When using the reduced graph, we differentiate between heuristic link assessment (using Algorithm 1) and an exhaustive search. When using heuristic link assessment in the reduced graph, we will follow the steps on the left in Figure 5.8. For exhaustive link assessment in the reduced graph, the procedure on the right is executed, nonetheless only links complying to the conditions of the reduced graph are allowed.

Next to the reduction in number of links by each of the methods, the percentage of links that are in the resulting graphs out of the links selected by the optimal outcome (the mathematical solver in Figure 5.8) is examined. This is done in order to show performance degradation, since the optimal links will not be a viable solution for the heuristic method. The analysis is done for 1,000 snapshots from [34] with 100 ms separation, using five nodes and three intended receivers. The northernmost node was assigned the transmitter and three of the remaining nodes were randomly assigned to be intended receiver. See Table 7.1 for the results.

Table 7.1: Evaluation of the link assessment and graph reduction steps.

Link assessment	**Reduced graph**	**Reduction in #links (%)**	**Detected links (%)**
Heuristic	No	34.68	84.01
Exhaustive	Yes, option 1	60.65	99.86
Exhaustive	Yes, option 2	72.35	95.47
Heuristic	Yes, option 1	69.82	83.53
Heuristic	Yes, option 2	76.25	81.86

For reference, the results from Section 6.3 are given in the first row.

It can be seen in the second row that with exhaustive link assessment, reducing the graph using option 1 results in a significant reduction of the number of links, whereas it still includes almost all links that were selected by the optimal solution. Using option 2 as shown in the third row, the reduction is around 19% more, but also 4.3% less links are detected.

Using heuristic link assessment, the reduction in number of links is even more, but not to the extent from Section 6.3. This means that from the links that are lost due to heuristic link assessment, a significant part would also not be included in the reduced graph. It can also be seen that the percentage of detected links for all links is 84.01%, while using the reduced graph it is only slightly lower, namely 83.53% for option 1 and 81.86% for option 2. The difference in terms of links reduction between option 1 and 2 for heuristic link assessment is 9%, resulting in 2% less links that are correctly included.

In conclusion, we reduced the graph resulting from a topology by formulating conditions that state whether a node or link should be included, based on the intended receivers that the nodes can serve. For a multicast link, two options were presented, of which the latter is more strict. From the results, we conclude that the graph reduction step leads to a significant reduction in the number of links in the graph, while it does not perform much worse, especially as compared to the result after heuristic link assessment. Furthermore, option 2 leads to a limited additional number of non-detected links, while it reduces the amount of links in the graph significantly. So, the graph reduction step with option 2, i.e. Constraint 7.3, will be used in the remainder of this thesis.

Chapter 8

Transmission tree generation

Now that the reduced graph has been set up, a tree should be created which determines the transmitter and receiver sets to choose in order to reach the intended receivers. This tree will be called the transmission tree $\mathcal{T}$ from now on. At first, a heuristic that does not yet take scheduling into account, i.e., solely the minimum Steiner arborescence in directed hypergraphs, is considered. Secondly, the joint problem of routing and scheduling is addressed by creating a transmission tree in which the order of inserting transmissions determines the schedule. Several options to select multicast transmissions will be presented. Each of the different methods we propose are evaluated for their performance. The algorithms described in this chapter are meant to be executed by the origin node at the message generation time and do not yet consider multiple active messages. The distributed version which allows for multiple messages is presented in Chapter 10.

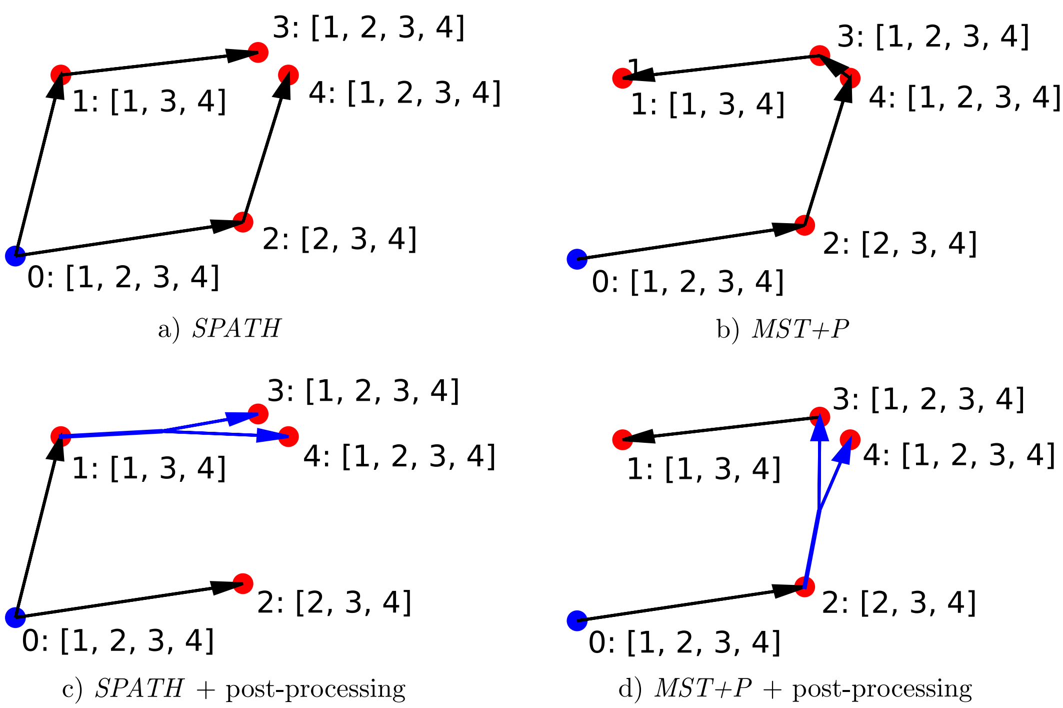
8.1 Minimum Steiner arborescence

There exist several heuristics for solving the minimum Steiner arborescence problem in regular graphs, of which two relatively simple ones from [36] are considered. This first one is called *SPATH*: shortest paths. This algorithm starts with a tree consisting of the root vertex (origin node) only and adds every edge and vertex on the shortest path from the root vertex to a terminal (intended receiver) to this tree. This is done for all terminals.
The second heuristic is called *MST+P*, which stands for minimum spanning tree and pruning. As the name suggests, it calculates the minimum spanning tree (arborescence) for all vertices in the graph and afterwards removes all non-terminal vertices and their incoming edges that are not leaves.
Next, we will consider an approach to apply these heuristics in hypergraphs, i.e. to allow multicast links in the graph and determine when to use these.

8.1.1 Converting to hypergraphs

The first approach we will consider to solve the minimum Steiner arborescence in **hyper**graphs is to solve the problem using one of the heuristics in a regular graph using single edges only, and afterwards replacing single edges with hyperedges. We will call this approach

post-processing from now on.
In the considered application there will always be single edges (unicast links) to the head vertices of a hyperedge (multicast link). Therefore, the connectivity of the regular graph is the same as the hypergraph and thus no nodes are disconnected when the regular graph is considered only.
Both trees resulting from the heuristics *SPATH* and *MST+P* can be improved by replacing multiple single edges by a single hyperedge if the cost of the hyperedge is lower than the sum of costs of the single edges. Since a vertex might have multiple incoming hyperedges, but only one incoming edge per vertex is needed, one hyperedge to be added to the tree should be selected. One way to do this is by traversing the created tree, either by a bread-first or depth-first search. When encountering a vertex that has an outgoing hyperedge in the hypergraph, the total cost of reaching the head vertices (receivers) of this hyperedge via single edges only is calculated. If the vertex has multiple outgoing hyperedges, the calculation is done for all of them. The hyperedge that has the lowest cost is chosen, provided that it is lower than the sum of single edges.
For an example graph, the trees resulting from both algorithms and applying *post-processing* on them is shown in Figure 8.1.



a) *SPATH* b) *MST+P*

c) *SPATH* + post-processing d) *MST+P* + post-processing

Figure 8.1: Trees resulting from SPATH (a), MST+P (b) and their post-processing variants (c and d).

It can be seen that the two heuristics result in different graphs and their performance will differ per scenario. For example, transmissions resulting from the tree of *SPATH* can be scheduled concurrently, whereas those resulting from *MST+P* should be sent consecutively.
Post-processing guarantees to provide a Steiner arborescence in the hypergraph that has a lower cost than the Steiner arborescence of the regular graph. However, when evaluating the

delay induced by all consecutive edges that are taken, this does not hold anymore. This is because the additional delay of a multicast link will be the same for the transmitter and all receivers of the multicast link. However, for a unicast link, the delay of only two nodes is affected.

Furthermore, *post-processing* can be considered as a rather simple approach, because it does not dynamically adapt the graph based on the nodes visited. In this way, while creating the tree, it does not take advantage of the fact that multicast receivers can be reached with lower cost in total.

To address these shortcomings, we will introduce a different approach using a constrained Steiner tree abstraction, which is explained hereafter.

8.2 Constrained Steiner tree

This second approach will consider the complete problem jointly by modifying the heuristic for the constrained Steiner tree problem of [33]. As explained in Chapter 5, the heuristic finds a minimum Steiner tree in case a delay bound is specified. The heuristic is suited for unicast transmissions only, but we will extend it in Section 8.3 to also support multicast transmissions. Although it is shown that other heuristic methods based on a tabu-search approach from [37] and a Greedy Randomized Adaptive Search Procedure from [38] perform better, these methods are non-deterministic, because they start with a random initial solution and proceed searching for better solutions from there. This forms a problem in distributed systems, as the nodes should agree on a solution together. Furthermore, these heuristics will produce better results depending on the number of iterations that are done, making it hard to evaluate their performance given that it should operate in a real-time system.

Algorithm description

In short, the heuristic from [33] works as follows. At first, the shortest paths between all pairs of nodes in the graph $\mathcal{G}(\mathcal{MPR}, \mathcal{E})$ are calculated, provided that the delay of this shortest path is less than the delay bound O. The shortest path includes which nodes need to transmit to reach R from T and is given by $\mathcal{P}(T \rightarrow R)$. The cost of the shortest path from T to R is given by $\mathcal{P}_C(T \rightarrow R)$, which is equal to the sum of edge costs (transmission durations) on this path. After the shortest paths are determined, a tree $\mathcal{T}$ will be built greedily by determining which nodes to visit. In the beginning, only one node (the origin node) is part of the visited nodes $\mathcal{V}$. The edges and vertices of the shortest path from a visited node to a terminal (intended receiver) for which the cost function f_{cost} has the lowest value are added to the tree. In this way, the number of visited nodes increases, whereas the number of nodes still to find decreases step by step. Adding new paths to the tree is done until all terminal nodes are covered.

The first adaptation to this heuristic that we propose is to incorporate scheduling specific for directional antennas, such that spatial sharing can be exploited. We will explain this adaptation in Section 8.2.1 for unicast transmissions only and it is extended for multicast transmissions in Section 8.3.

8.2.1 Directional constrained Steiner tree

In [33], two different cost functions are given. The output of both cost functions is infinite if the sum of the path delay from the origin node to the visited node and the additional delay of the shortest path is higher than the delay bound. However, in case of directional antennas it is important in which order transmissions by the same node are processed. Therefore, the order in which transmissions are added to the tree determines the schedule that has to be followed. The modifications that we suggest are specific for these type of systems and the resulting algorithm will be further referred to as the *Directional Constrained Steiner Tree (DCST)* heuristic. Algorithm 3 presents the pseudocode of *DCST* for unicast transmissions. Instead of the path delay to a specific node, the delay of that node in the transmission tree that is set up so far is used. The delay of node n in the transmission tree is given by $D_{\mathcal{T}}(n)$ and changes while building the tree. For a specific edge to be included in the tree, only the delay of the nodes participating in the transmission which is represented by this edge is updated. Specifically, the updated delay is equal to the current delay of the transmitter, summed with the duration of the transmission. The updated delays for a transmission from T to R having transmission tree $\mathcal{T}$ are given by:

$$D'_{\mathcal{T}}(T) = D'_{\mathcal{T}}(R) = D_{\mathcal{T}}(T) + E_C(T \rightarrow R) \tag{8.1}$$

As an example, consider the steps 1-4 of creating a transmission tree and its corresponding schedule in Figure 8.2.

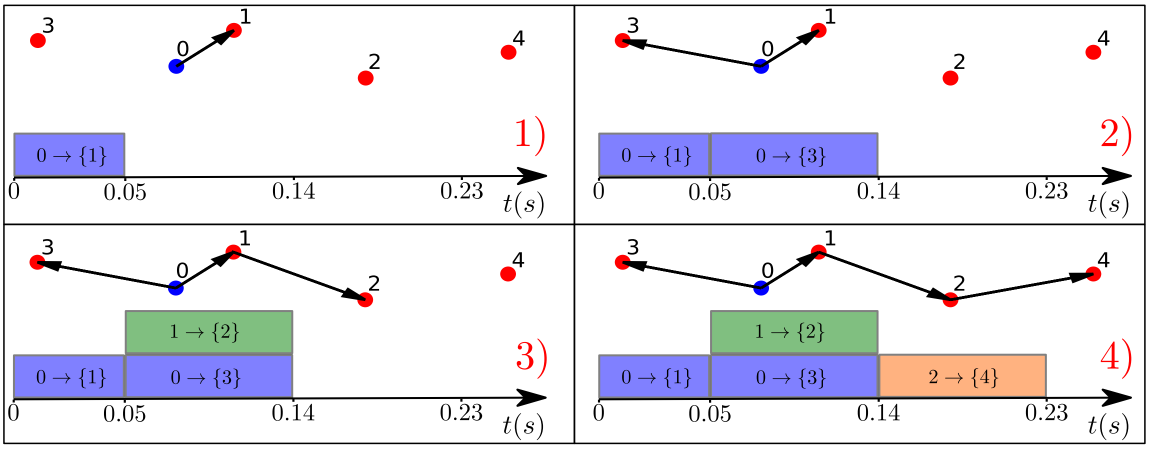


Figure 8.2: Transmission tree with its corresponding schedule.

Let the message generation time be equal to $t = 0$. Then, the delay of node 0 and 1 equals 0.05 s after the first step (top left). After the second step (top right), the delay of 1 is still 0.05 s, whereas that of 0 and 3 now equals 0.14 s. Only the delays of 1 and 2 are updated to 0.14 s after the third step (bottom left). Finally, after the fourth step (bottom right), $D_{\mathcal{T}}(2) = D_{\mathcal{T}}(4) = 0.23$ s.

If a shortest path to add consists of multiple edges, the delays for each of the participating nodes are updated subsequently. This is described in lines 16-17 in Algorithm 3. In order for a new path to be included in the tree, the updated delay of the intended receiver to be included should be lower than the timeout. The cost function will take this into account, which will be elaborated on hereafter.

Cost function

Several cost functions can be used to determine the order in which intended receivers will be handled and by which visited node. The cost will be infinite for all cost functions if $D'_{\mathcal{T}}(R)$ is higher than the timeout. We will refer to the value a cost function takes when this is not the case as $C_{\mathcal{P}}$. This value varies for the different functions. Six different cost functions will be presented here and evaluated afterwards.

Cost functions f_C and f_{CD} are adopted from the constrained Steiner tree heuristic from [33]. The simplest cost function f_C uses $C_{\mathcal{P}}$ equal to the cost of the shortest path from visited node T to the intended receiver R, which is the sum of transmission durations along this path. The function f_C is given by:

$$f_C = \begin{cases} \mathcal{P}_C(T \to R) & \text{if } D'_{\mathcal{T}}(R) \leq O \\ \infty & \text{otherwise.} \end{cases} \tag{8.2}$$

In order to increase the chance to extend the path even after the chosen node, another cost function f_{CD} incorporates the residual time after this path as well. This is given by:

$$f_{CD} = \begin{cases} \frac{\mathcal{P}_C(T \to R)}{O - D'_{\mathcal{T}}(R)} & \text{if } D'_{\mathcal{T}}(R) \leq O \\ \infty & \text{otherwise.} \end{cases} \tag{8.3}$$

When comparing f_C with f_{CD}, a couple phenomena become clear. Function f_{CD} often has the tendency to spread the load across the nodes better, because it also bases the choice on the residual time, which will be higher for nodes that participated less in transmissions. However, in the case the load on the nodes does not yet cause a problem, it does not follow the most logical path, which might result into a higher total delay.

Next, to exploit spatial sharing, we present a simple, yet more interesting cost function for the directional constrained Steiner tree heuristic, called f_D. In this function, the value $C_{\mathcal{P}}$ is the updated delay of the intended receiver that is chosen, and it is given by:

$$f_D = \begin{cases} D'_{\mathcal{T}}(R) & \text{if } D'_{\mathcal{T}}(R) \leq O \\ \infty & \text{otherwise.} \end{cases} \tag{8.4}$$

In this way, the cost is still implicitly determined by the additional transmission duration, but it prefers to use receivers and transmitters that do not have a high delay yet. In a way, function f_{CD} does the same, but it tries to address the trade-off between path cost and delay, whereas in case the path cost is equal to the transmission duration, a transmission resulting in the minimum delay should always be favored. Therefore, f_D is more suited for a system using directional antennas, having as objective to reach the maximum number of intended receivers before the timeout. We will consider an additional factor for these cost functions next, to influence the order of building the tree.

Algorithm 3: Directional constrained Steiner tree algorithm (unicast only).

Input : $\mathcal{G}(\mathcal{MPR}, \mathcal{E})$, n_{m_j}, $\mathcal{R}_{m_j}$, O
Output: Transmission tree $\mathcal{T}$.

begin
  foreach $T, R \in \mathcal{MPR}$ **do**
    Compute shortest path provided that it is within the timeout.
    $P(T \to R) \longleftarrow$ Nodes along shortest path.
    $P_C(T \to R) \longleftarrow$ Sum of transmission durations along the path.
  $\mathcal{V} \longleftarrow \{n_{m_j}\}$ `/* Already visited nodes. */`
  $\mathcal{U} \longleftarrow \mathcal{R}_{m_j}$ `/* Nodes still to find. */`
  $\mathcal{T} \longleftarrow \emptyset$ `/* Transmission tree. */`
  foreach $n \in \mathcal{N}$ **do**
    $D_{\mathcal{T}}(n) \longleftarrow 0$ `/* Delay of node` n `in tree. */`
  while $\mathcal{U} \neq \emptyset$ **do**
    $minCost = \infty$
    foreach $T \in \mathcal{V}$ **do**
      foreach $R \in \mathcal{U}$ **do**
        if $\mathcal{P}(T \to R)$ *exists* **then**
          `/* Update delays for each Tx-Rx pair in path. */`
          foreach $n_1, n_2 \in \mathcal{P}(T \to R)$ **do**
            $D'_{\mathcal{T}}(n_1), D'_{\mathcal{T}}(n_2) \longleftarrow D_{\mathcal{T}}(n_1) + E_C(n_1 \to n_2)$
          `/* Evaluate cost of path. */`
          $cost \longleftarrow f_{cost}(\mathcal{P}_C(T \to R), D'_{\mathcal{T}}(R))$
          if $cost < minCost$ **then**
            $nextPath \longleftarrow \mathcal{P}(T \to R)$
            $minCost \longleftarrow cost$
            Save updated delays $D'_{\mathcal{T}}$ for nodes in $nextPath$.
    if $minCost == \infty$ **then** `/* No path found within timeout. */`
      return $\mathcal{T}$
    $\mathcal{T} \longleftarrow \mathcal{T} + nextPath$ `/* Add path to the tree. */`
    $\mathcal{V} \longleftarrow \mathcal{V} \cup \{n \mid n \in nextPath\}$ `/* Visit all nodes along path. */`
    $\mathcal{U} \longleftarrow \mathcal{U} \setminus \{R \mid R \in nextPath\}$ `/* Remove intended receiver. */`
    foreach $n \in nextPath$ **do**
      $D_{\mathcal{T}}(n) \longleftarrow D'_{\mathcal{T}}(n)$ `/* Update delay. */`
  return $\mathcal{T}$

Order of transmissions

As explained before, when using directional antennas the order of transmissions influences the result, since it determines which transmissions can be sent concurrently. Furthermore, it determines which nodes receive the message first and therefore also which are favorable for relaying the message later on. Therefore, a more sophisticated cost function influencing this order is considered next. It reasons from the fact that if the tree builds first towards the direction where more MPRDs can be handled, the chance is higher that more receivers will be reached within the timeout. For example, consider the topology in Figure 8.3.

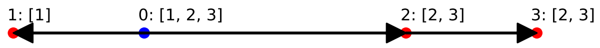


Figure 8.3: Topology where the receivers have a different number of MPRDs.

Although the delay for sending towards node 1 is lower, it is beneficial to send to node 2 first, because after node 2 receives the message, it can forward it to 3. The transmission from 2 to 3 and from 0 to 1 will then happen simultaneously and the total delay is lower.

The difference between nodes 1 and 2 in this topology is that 2 has more MPRDs that it can handle. So, when the algorithm evaluates which receiver to address next, it is wise to incorporate the number of MPRDs that it can handle. Therefore, we will introduce an additional factor to the value of $C_{\mathcal{P}}$ in the cost functions:

$$\frac{1}{|MPRDs_r(R)| + 1}, \tag{8.5}$$

where $|MPRDs_r(R)|$ is the number of MPRDs of the intended receiver that are not yet addressed, even after this transmission. Thus, a path to an intended receiver is decreased if it can still handle MPRDs once it received the message. The term $+1$ is introduced such that if a node has no MPRDs left that still need to be handled, the cost simply equals $C_{\mathcal{P}}$. For two paths with a difference of 1 between their values of $|MPRDs_r(R)|$, the influence is larger if the value $|MPRDs_r(R)|$ itself is smaller, due to the use of a fraction. Thus, there is a stronger tendency to favor one with more MPRDs if in total less MPRDs are to be handled. This is reasonable, since if both can still handle multiple MPRDs, the benefit of choosing one over the other is less.

Cost functions that use the factor of Equation 8.5 are denoted by the additional letter M from now on. In total, we can now formulate six cost functions, namely f_C, f_{CD}, f_D, f_{CM}, f_{CDM} and f_{DM}. Next, we will consider a slightly different factor to unburden transmitters.

Unburden transmitters

The cost functions can be improved in some cases by taking the number of MPRDs of the transmitter into account. Namely, if there are multiple transmitters that can reach a receiver, it should be handled by the one with the least MPRDs left, such that the others have more time to handle the remaining MPRDs.

Let $MPRDs_r(T)$ be the remaining MPRDs of the transmitter, even after the considered transmission has been successfully transmitted. Then, the cost functions can be modified by multiplying $C_{\mathcal{P}}$ with the factor:

$$\frac{|MPRDs_r(T)| + 1}{|MPRDs_r(R)| + 1}, \tag{8.6}$$

instead of the factor from Equation 8.5. In this way, a transmitter with less MPRDs left and a receiver with more MPRDs left are favored jointly. In a way, cost functions f_D and f_{DM} provide this already, since if multiple transmitters can reach a receiver, one with lower delay is favored.

On the other hand, this factor might work counterproductive in some scenarios. This is especially the case when the transmission tree resulting from cost functions without this ratio has only one branch, which is usually the situation in the *Line* network. Namely, including this ratio might lead to a transmitter sending twice more often, if it has few MPRDs to handle. In the *Line* network, this is not beneficial, because the message has to be forwarded in one direction, such that the most recent receiver is usually favored for relaying. On the other hand, the number of branches of a transmission tree can be predicted by looking at the number of completely disjoint MPR sets, meaning that none of the MPRDs of a set of MPRs are an MPRD for any of the MPRs of another set. A cost function can be chosen based on this number, however since by using f_{DM} transmitters that are less burdened are chosen inherently, this is not worked out further. Hereafter, we will evaluate the six cost functions presented before.

Evaluation

The different cost functions are evaluated by generating 5,000 artificial scenarios with 10 nodes and 5 intended receivers. One message is generated with a timeout of 0.5 s. Two different networks are considered, which we call the *Square* and *Line* network. In the *Square* network, the nodes are randomly placed in a square area of 350×350m. The origin node is placed in the middle of this area. In the *Line* network, the nodes are randomly placed in an area of 14×700m, in order to simulate a highway scenario. The origin node is placed on one end of the vertical lane.

For each random scenario, the reduced graph is created by following Algorithm 2, however only for unicast links. Then, Algorithm 3 is applied using different cost functions. Two different metrics are considered; the first one is the number of intended receivers that are reached before the timeout according to the created transmission trees. If these are the same for all cost functions, the total delay in order to reach these intended receivers is evaluated as well. This is given by the time between message generation and the time at which the last intended receiver successfully received the message.

For the *Square* network, in 86% of the scenarios, the number of intended receivers reached is the same for all cost functions. The average delay and average number of intended receivers reached before the timeout for this network is shown in Figure 8.4. For visibility, the y-axes in these graphs do not start at 0.

It is clear that including the ratio of Equation 8.5 results into better performance for all three cost functions. Furthermore, it can be seen that the delay when using f_C or f_{CM} compared to the others is higher, but it reaches slightly more intended receivers. Moreover, f_{CD} and

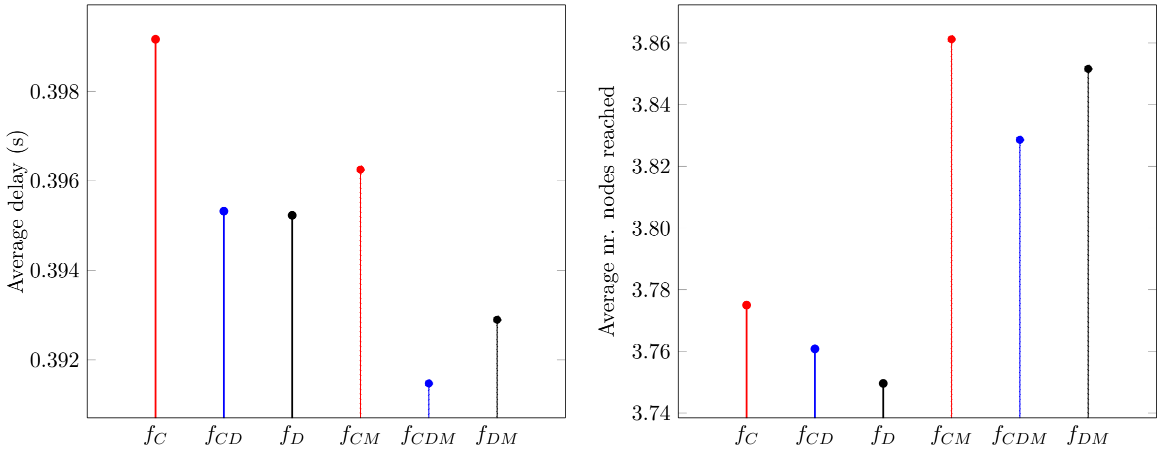


Figure 8.4: Average delay (left) and average number of intended receivers reached before the timeout (right) for different cost functions considering the Square network.

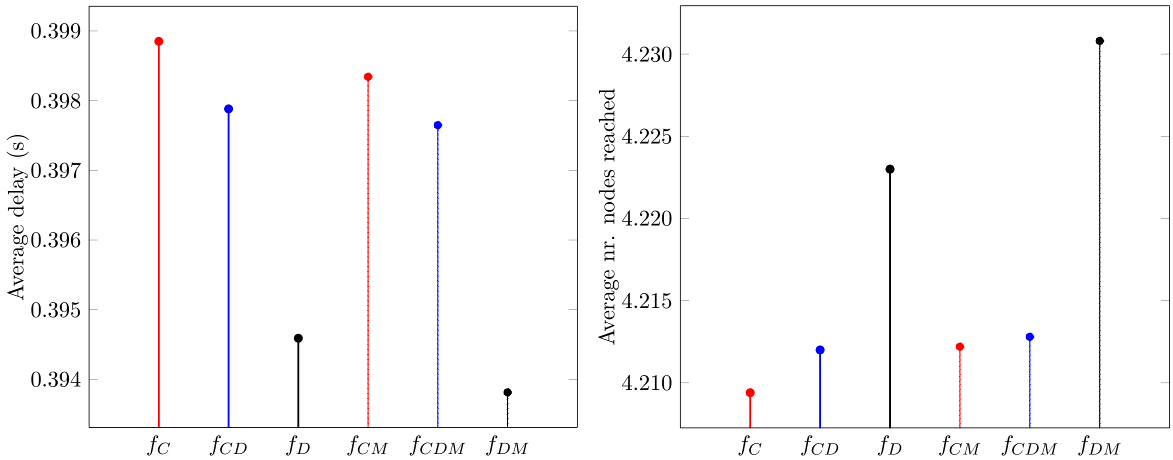


Figure 8.5: Average delay (left) and average number of intended receivers reached before the timeout (right) for different cost functions considering the Line network.

f_D as well as f_{CDM} and f_{DM} perform roughly equally well. Thus, for the *Square* network, either of f_{CM}, f_{CDM} or f_{DM} can be used with roughly equal performance.

In the *Line* network, the different cost functions result in the same number of intended receivers reached in 97% of the cases, which is even higher than for the *Square* network. This is mostly caused due to the fact that performance improvement by including the ratio of Equation 8.5 is less significant compared to the *Square* network. This can be seen in Figure 8.5. It can be explained by the fact that the impact of the order of transmissions is less, since the number of MPRDs among the receivers does not differ significantly.
Furthermore, it becomes clear that f_D and f_{DM} perform significantly better than the others. This is because the benefit of concurrent transmissions is better exploited in the *Line* network. Thus, overall the best choice seems to use cost function f_{DM}.

To conclude, we adapted the constrained Steiner tree heuristic of [33] such that it can be applied to systems with directional antennas. We examined two existing cost functions and we designed a cost function f_D that suits better for directional antennas, as it incorporates the delay directly. These three cost functions were extended to take the order of transmissions into consideration. We showed that the cost function f_{DM} has the overall better performance and decided that it will be used for the remainder of this thesis. Next, we introduce three methods to extend the directional Steiner tree algorithm in order to support multicast transmissions.

8.3 Multicast selection

The shortest path to a single node as calculated in lines 1-4 of Algorithm 3 will usually not include multicast transmissions, since the cost of a multicast is at least as high as a unicast link. In order to make use of multicast transmissions, their benefit should be assessed in a different way. A multicast transmission is only favorable when reaching the receivers in one transmission results into less delay than reaching them via unicast transmissions. In order to address this in the most accurate way, a condition we use is that a multicast link should have a lower cost (transmission duration) than the cost of the minimum spanning tree formed by unicast links between the transmitter and receivers of the multicast link. Multicasts for which this holds are called *relevant multicasts* from now on. For example, consider the multicast link on the left in Figure 8.6 and the minimum spanning tree formed by unicast links on the right.

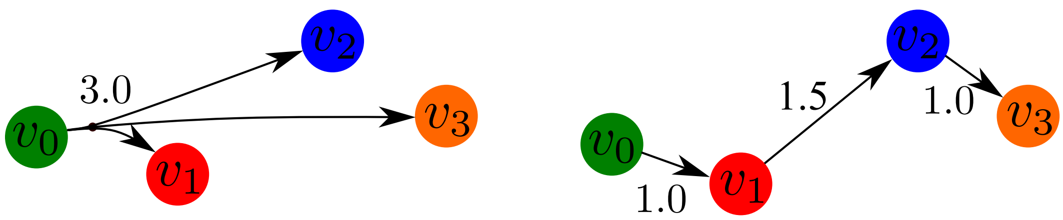


Figure 8.6: Relevant multicast (left) and the minimum spanning tree formed by unicast links (right).

Since the cost of the multicast link (3.0) is lower than the cost of the minimum spanning tree (which is 3.5), the multicast link is a relevant multicast. This definition will be used in the methods we describe hereafter.

8.3.1 Description of several methods

We introduce three methods to evaluate whether a multicast should be included in the transmission tree: *stepwise*, *average cost* and *post-processing*. Each method will be discussed hereafter.

Stepwise

At first, a rather free approach will be considered, which we will refer to as *stepwise*. At each step in Algorithm 3 (lines 12-29), first the unicast choice will be determined. Afterwards, it will be evaluated if there is a relevant multicast transmission that includes this unicast receiver. If there exists one, it will be used instead of the unicast link. If there are multiple relevant multicasts available, the one that covers the most receivers will be chosen. When there are still multiple available, the path with the lowest cost (using f_{DM}) is chosen. For the value of $|MPRDs_r(R)|$ in Equation 8.5, only the distinct MPRDs of the complete receiver set are counted. For the remainder of the steps in the algorithm, multicast links that include at least one already visited node are not considered further on.

On the one hand, this method might lead to including relayers at relatively low additional cost, which can be helpful in the tree later on. On the other hand, when nodes are included in a multicast (which has some additional cost as compared to a single unicast) that do not contribute to the tree, the overall performance is worse. Although the reduced graph already filters links that are certainly not relevant, it is not trivial to predict whether including a specific receiver set will contribute to the tree in a positive way. Obviously, when all receivers are intended receivers, this is the case. However, non-intended receivers addressed in a multicast should be used to relay the message later on. Whether they will be selected as transmitter depends on several factors, such as its path loss to MPRDs or other MPRs, compared to other nodes in its neighborhood that have the message. Yet also its delay compared to these nodes is important.

Besides, since nodes are half-duplex, the transmitter and all receivers are delayed with the duration of a multicast, whereas when this transmission is instead split into multiple unicasts, the first transmitter(s) will experience less delay. Therefore, in a multicast some of the nodes are kept busy for longer, thereby possibly postponing the delivery of other transmissions, which might affect the number of reached intended receivers. Deciding which multicasts are likely to be useful and thus may be included in the tree will be addressed in the next two proposals: *average cost* and *post-processing*.

Average cost

For the second approach, instead of only single nodes, the receiver sets of relevant multicasts are also included in the set of nodes still to visit ($\mathcal{U}$). The algorithm operates in the same way, except that in order to let multicasts compete with unicasts, inside the cost function the duration of a multicast transmission is divided by the number of receivers addressed. In this way, an average cost is used, similar to the average broadcast time used in [26]. If a multicast transmission is included in this way, all links that include one of its receivers are now excluded from $\mathcal{U}$.

This rather simple approach ensures that multicasts are favored less often if they have a relatively high cost compared to a unicast. However, with this approach the full potential of multicasts is not exploited, since even if the average cost is not lower than a unicast, using a multicast transmission might be beneficial.

Next to this, there are two other difficulties regarding the assessment of using a multicast. The first one is that even if the duration of the multicast is less than the total duration of the unicast spanning tree covering the multicast receivers, the unicast alternative might still result in a maximum delay that is lower than caused by the multicast. This is when concurrent transmissions that make use of the fact that only two nodes participate in a unicast can take place. This often happens when using cost function f_D or f_{DM}.

One way to address these additional difficulties is to use the information about the unicast alternative. This can be achieved using the method *post-processing*.

Post-processing
The last method follows the *post-processing* approach as explained in Section 8.1.1 in the sense that the directional constrained Steiner tree heuristic is first evaluated with unicast transmissions only. Afterwards, unicast transmissions can be replaced by multicast transmissions if the multicast receivers are used in the unicast tree. Yet, it should be evaluated first whether at least one receiver receives the message earlier than in the unicast alternative. Only then, the multicast is potentially beneficial for the tree. Besides, transmissions that get delayed due to the multicast transmission should not violate the timeout. This is checked by inserting the multicast transmission in the already created unicast schedule. If a transmitter can send multiple multicast transmissions for which these two conditions hold, the one with the largest receiver set is chosen. When there are still multiple available, the one with lowest cost is chosen.
In this way, it is guaranteed that the number of intended receivers reached before the timeout by using a multicast is equal to or greater than that of the full unicast tree. However, in order to check these conditions, alternative schedules using multicasts should be set up, which leads to additional computations. Furthermore, in order to make this work, the algorithm should build the unicast tree even beyond the timeout, such that when the tree can grow larger due to the multicast transmissions, there are still unicast links to replace.
As previously mentioned, the downside of the *post-processing* approach is that it does not take the full advantage of reaching multiple receivers at once, as the structure of the tree remains unchanged. It does not include possible relayers if those are not used in the unicast tree. Furthermore, some of the unicast relayers might become redundant using multicast transmissions, but they will be included in the post-processed tree.

8.4 Evaluation

In this section, we provide a performance evaluation via simulations. The goal of this analysis is threefold. First, we want to evaluate the performance of *post-processing* applied on the trees resulting from the heuristics *SPATH* and *MST+P*, as well as on the tree created by *DCST*. Secondly, we want to investigate the impact of the network size in realistic scenarios on the performance of *DCST* using unicast links and with the multicast selection methods. Lastly, we want to evaluate the computational complexity of *DCST* when using only unicast links, as well as for each of the multicast methods. Furthermore, we are interested in the computation time of transmission tree generation by using the reduced graph as compared to using all links.

Performance of *DCST*
Firstly, we will evaluate the benefit of using *post-processing* after the directional constrained Steiner tree heuristic (*DCST*) as compared to using it after the simple heuristics *SPATH* and

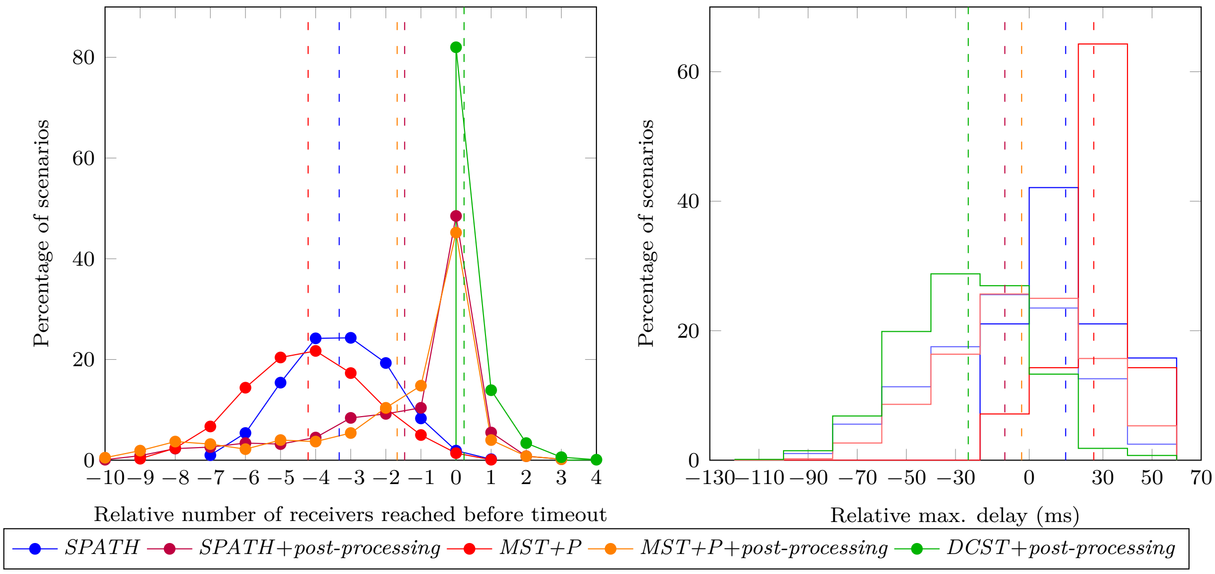


Figure 8.7: Distribution of the relative number of intended receivers reached before the timeout as compared to unicast for the different methods.

Figure 8.8: Distribution of the relative maximum delay as compared to unicast when the number of intended receivers reached is the same.

MST+P. For *SPATH* and *MST+P*, the schedule is derived from their trees in the same way as by using cost function f_{DM}. That is, for each of the links in the trees, the resulting delay and the value of the factor from Equation 8.5 is determined, which can be used to determine the transmission with the lowest cost. This transmission is added to the schedule, until no transmission can be scheduled anymore before the timeout. As input, we will use the reduced graph using option 2 (the procedure on the left in Figure 5.8).

In total, we use 1,000 snapshots with 100 ms separation of the five-lane US Highway 101 from [34], as elaborated on in Section 6.3. From the snapshots, we select a segment containing 15 nodes. We assign the northernmost node as the origin node, while 10 of the remaining nodes are randomly assigned as intended receivers. The timeout was set to 250 ms, leading to reaching all intended receivers in time by *DCST* in 71.9% of the scenarios.

To understand the benefit of multicast transmissions, we analyze first the relative number of intended receivers reached before the timeout, as compared to *DCST* using unicast links only. The distribution for the different scenarios is shown in Figure 8.7. Furthermore, we will consider the maximum delay, meaning the time between message generation and the time at which the last intended receiver received the message. The relative value as compared to using *DCST* with unicast links is evaluated, but only in case the number of intended receivers reached is the same. The distribution of the relative maximum delay for the different scenarios is shown in Figure 8.8. In both figures, the average values for each method are shown with dashed vertical lines.

With respect to the unicast links only, it can be seen that both *SPATH* and *MST+P* perform significantly worse as compared to *DCST* (the baseline). Both methods often reach 2 to 5 intended receivers less before the timeout. The maximum delay is on average 14.8 ms higher for *SPATH* and even 26.3 ms higher for *MST+P*. Especially the latter does not perform well, as it does not often result into concurrent transmissions in the schedule.

When *post-processing* is applied, all methods are significantly improved. As can be seen in Figure 8.7, *DCST+post-processing* always reaches 0-4 intended receivers more than *DCST*. *SPATH+post-processing* and *MST+P+post-processing* still reach significantly less intended receivers on average than *DCST+post-processing*. Also in terms of maximum delay, these methods perform worse than *DCST+post-processing*. Thus, indeed *DCST* greatly outperforms the simple heuristics, regardless of whether *post-processing* is used.

Performance of multicast selection methods as compared to *unicast*

Next, the performance of the three options for selecting multicast links using *DCST* is compared with *DCST* using unicast links only, referred to as *unicast*. For this, we derive topologies from the dataset of [34]. For each of the methods *unicast*, *stepwise*, *average cost* and *post-processing*, the steps of generating the reduced graph using Algorithm 2 (using option 2) and creating a transmission tree using Algorithm 3 (using cost function f_{DM}) are executed, as shown on the left in Figure 5.8.

A scenario using 15 nodes and 10 intended receivers, which were randomly assigned, is considered. The origin node is located on one end of the highway segment. In total 1,000 snapshots were taken with 100 ms separation. The timeout was set rather tight, namely to 250 ms in order to make meeting the timeout for all intended receivers a challenge. Using *unicast*, in 72.6% of the cases, all intended receivers could be reached before the timeout.

In Figure 8.9, for each method the distribution of relative number of intended receivers reached before the timeout, as compared to the *unicast* method, is shown. The average relative number of intended receivers reached is shown with dashed vertical lines. It can be seen that the methods *average cost* and *post-processing* perform roughly equally well and on average better than *unicast*. The method *post-processing* performs, as expected, never worse than *unicast*, whereas *average cost* only in a very limited number of cases. The method *stepwise*, on the other hand, performs quite often worse than *unicast* and in not so many cases better.

In case there is no difference in the number of receivers reached, the distribution of maximum delay as compared to *unicast* is shown in Figure 8.10. It can be seen that *average cost* performs best here with an average difference of 29 ms, closely followed by *post-processing*. The method *stepwise* performs on average equally well as compared to *unicast*. However, all methods do perform worse in a notable number of scenarios. For *post-processing*, additional delay was allowed if it did not result into a violation of the timeout. This result shows that sacrificing faster delivery for some of the nodes by including them in a multicast is sometimes needed to obtain an overall better result.

Performance for different number and distribution of intended receivers

It is expected that the performance of the methods depends on the number of intended receivers, since the number of multicasts in the reduced graph depends on it. Next to that, if not all nodes are intended receivers, we will look at scenarios in which these are randomly

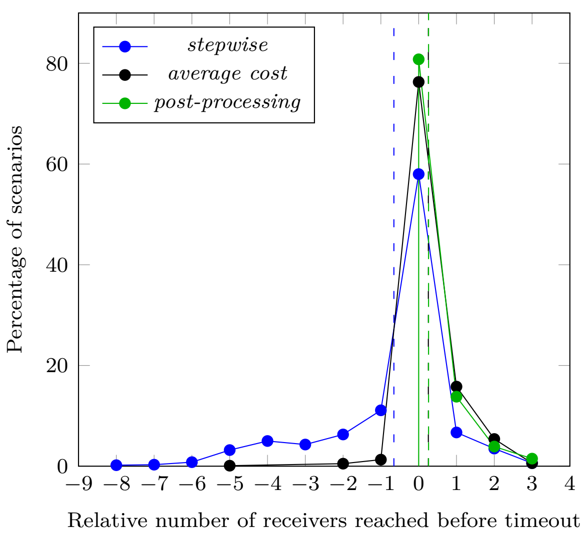


Figure 8.9: Distribution of the relative number of intended receivers reached before the timeout as compared to unicast for the different methods.

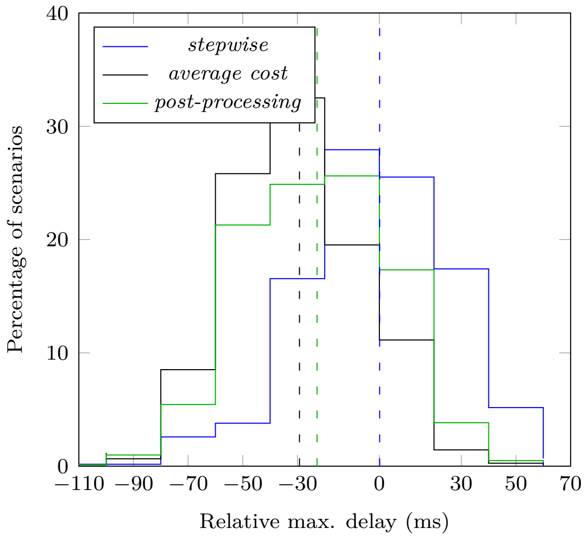


Figure 8.10: Distribution of the relative maximum delay as compared to unicast when the number of intended receivers reached is the same.

assigned and when they are located in a geocast area. For the latter case, all the intended receivers will be located on the opposite side of the segment from the origin node, such that the relaying nodes are in between. If all nodes are intended receivers, there is no difference between the two scenarios.

We will use 15 nodes for all snapshots, in which the northernmost node is assigned as the origin node. Either 5, 8, 11 or 14 nodes (all but the origin node) are assigned as intended receiver. To compare the different scenarios, the timeout was set arbitrarily high (10 s), such that all multicast methods reached the same amount of intended receivers as *unicast*. As metric, we will only use the average relative maximum delay as compared to *unicast*. The snapshots are now taken 20 s apart, such that the simulated scenarios can be considered independent. Due to the limited dataset size, we can evaluate 100 different snapshots, of which the average value for the metric is taken. See Figure 8.11 for the results including their 95% confidence intervals. Results for snapshots in which the intended receivers were randomly assigned are shown on the left, the ones in which the intended receivers are in a geocast area are on the right.

It becomes clear that the method *stepwise* behaves quite different as compared to *average cost* and *post-processing*, which show resemblance. For *stepwise*, there are cases when *unicast* is better on average, where this is not the case for *average cost* and *post-processing*. The average value for *stepwise* stays roughly the same in the random scenario, whereas in the geocast scenario there is slightly more variation. As can be seen from the confidence intervals, the more intended receivers, the variation per scenario increases. For the methods *average cost* and *post-processing*, on the other hand, the relative delay reduces with increasing number of intended receivers. This shows that these methods provide a consistent improvement as compared to *unicast*. It can be seen that for 5 intended receivers *post-processing* slightly

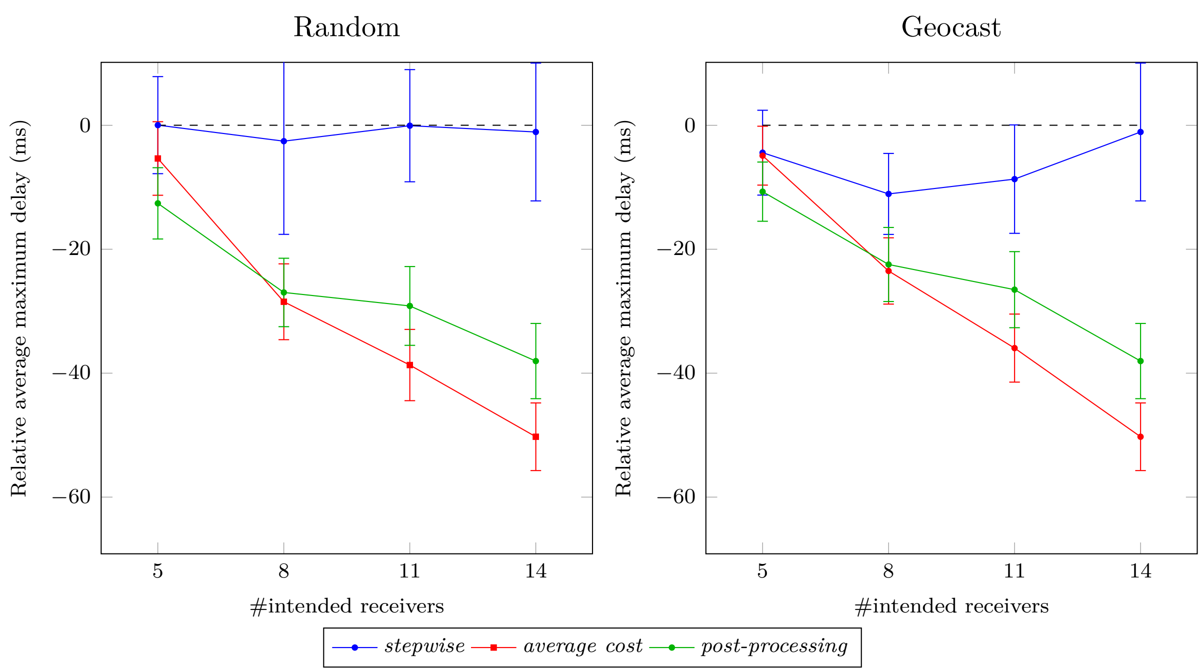


Figure 8.11: Average maximum delay as compared to unicast for different number of intended receivers, randomly distributed (left) or in a geocast (right).

outperforms *average cost* on average, where it is the other way around for a larger number of intended receivers.

The difference between random and geocast scenarios is not significant. The relative gap between *post-processing* and *unicast* for the random scenario ranges from 6.5% to 12.0% for 5 to 11 intended receivers. In the geocast scenario, the relative delay reduction is between 5.4% and 10.8%. When all nodes are intended receivers, the reduction is 14.6% (for both scenarios, since they are identical). For *average cost*, the relative delay reduction compared to *unicast* ranges from 2.8% to 16.0% in the random scenario and from 2.5% to 14.7% in case of a geocast. In case all nodes are intended receivers, the reduction is even 19.3%.

The performance can be better understood by looking at what kind of transmissions each method uses. First, we will look at the fraction of receivers that is handled in a multicast. Next to that, the fraction of intended receivers out of all receivers of the scheduled transmissions is an interesting metric. It shows to which extent non-intended receivers are used to relay the message. In case multicasts are used, non-intended receivers might be included that are eventually not used to relay. For the scenarios described above, the average fraction of receivers addressed via a multicast and the average fraction of receivers that are intended receivers are shown in Figure 8.12. The 95% confidence intervals are shown as well. Note that the y-axis in the right plot does not start at 0 for visibility.

In Figure 8.12 on the left it can be seen that especially for a smaller number of intended receivers, using *stepwise* most of the receivers are handled by a multicast, as follows from its loose approach. From 11 intended receivers on, for *post-processing* the fraction is roughly the same. Using *average cost* the least amount of receivers are reached via multicasts (except

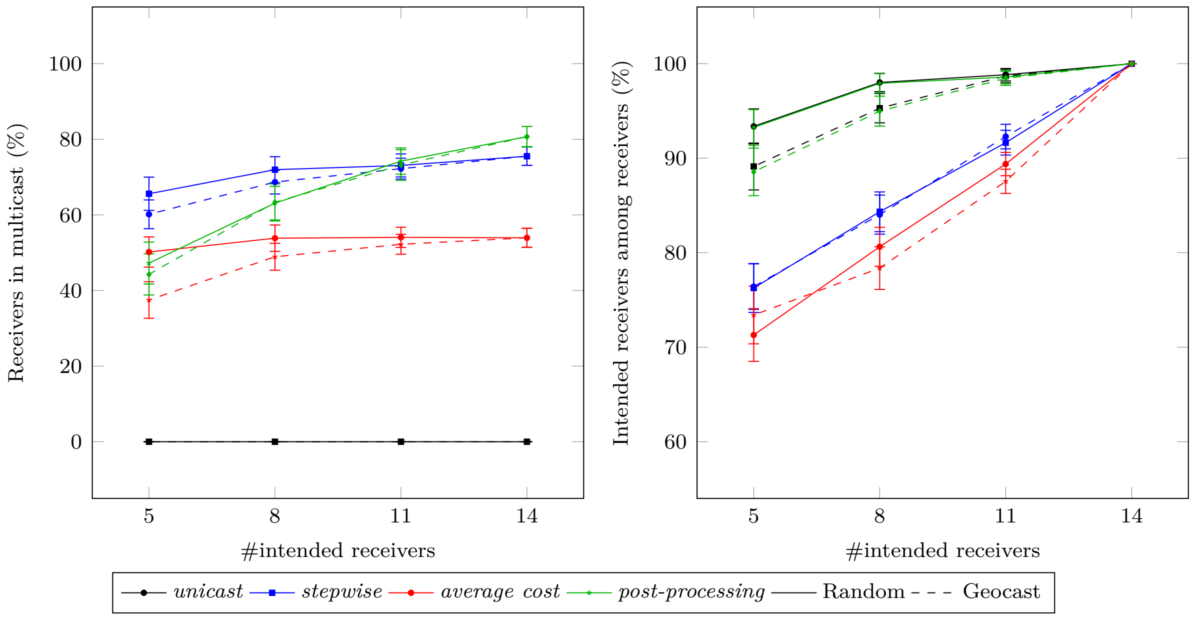


Figure 8.12: Average percentage of receivers addressed via a multicast (left) and percentage of receivers that are intended receivers (right).

for 5 intended receivers randomly distributed). In general, the more intended receivers, the higher the fraction, since also more multicast links are in the reduced graph. Furthermore, for randomly distributed intended receivers more receivers are handled by multicasts than for a geocast. This can be explained by the fact that in the area where only relay nodes reside, it is not likely that a multicast exist, since these nodes normally have the same MPRDs, namely the intended receivers in the geocast area.

Furthermore, the plot on the right in Figure 8.12 shows that the fraction of intended receivers for *post-processing* is high and almost the same as *unicast*, which is expected, as the procedure closely follows the *unicast* tree. For *post-processing* the fraction is only slightly higher, caused by trees in which already all intended receivers are reached where in the unicast case an additional relayer was used. *Post-processing* is followed by *stepwise*, because this method should choose multicasts that include the unicast choice. Due to its poor performance, it is expected that part of the non-intended receivers are included, while they are not used to relay. The method *average cost* includes relatively the most non-intended receivers, since its tree is built freely based on the new cost, not taking the unicast tree into account. As expected, the fraction increases with increasing number of intended receivers.

Thus, although *stepwise* has the highest fraction of receivers that are handled by multicasts, it does not perform well. The variation in performance per scenario can be explained by the fact that *stepwise* loosely chooses to use multicasts. As it does not consider the quality of this multicast, in some scenarios this is eventually beneficial for the tree, whereas in others it is not. The method *average cost* uses the least multicasts for reaching the receivers and includes the most non-intended receivers, though it performs roughly the same as compared to *post-processing*. This means *average cost* misses out on too many multicast options, since for it to include a multicast, the average cost should be lower than the least unicast cost. On the other hand, *post-processing* does not make full use of relaying nodes in multicasts, since it is tight to the unicast tree, as can be seen by the similar fraction of intended receivers among all receivers.

Performance for different number of nodes

Instead of an increasing number of intended receivers with a fixed amount of nodes, we will now look at scenarios with increasing number of nodes for a fixed percentage of intended receivers. We will use scenarios with 6, 11, 16 and 21 nodes, of which one is assigned the origin node. Each time 60% of the remaining nodes are assigned as an intended receiver. Again, we will consider both randomly assigned intended receivers and receivers in a geocast area. The snapshots are taken 20 s apart and the timeout was set to 10 s, such that all multicast methods reached the same amount of intended receivers as *unicast*. See Figure 8.13 for the average relative maximum delay as compared to *unicast* including the 95% confidence intervals, with the random scenario on the left and geocast on the right.

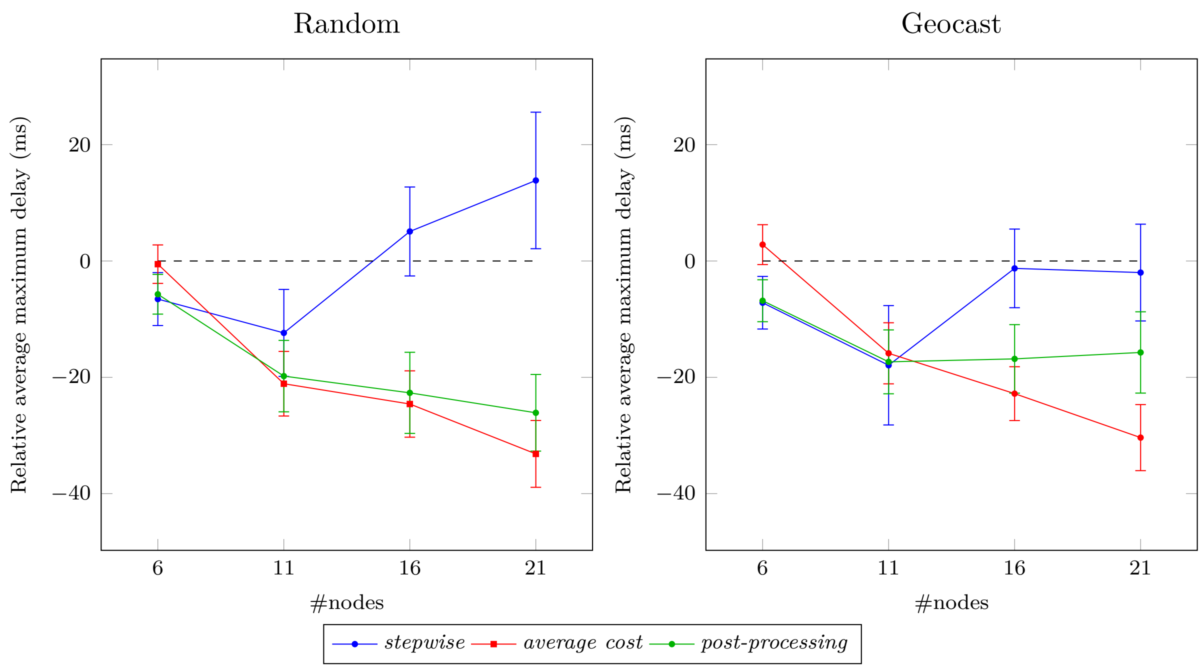


Figure 8.13: Average maximum delay as compared to unicast for different number of nodes. Randomly distributed intended receivers on the left, geocast on the right.

As shown before, *stepwise* behaves differently as compared to *average cost* and *post-processing*. Except for a geocast with 6 nodes, *average cost* performs on average better than *unicast*, which is also achieved by *post-processing*. *Stepwise* performs worse for scenarios larger than 11 nodes, whereas the latter methods show an increasing reduction for larger scenarios. Only in case of a geocast, *post-processing* stabilizes from 11 nodes on. Furthermore, *average cost* slightly outperforms *post-processing* for larger scenarios. To see how each of the methods approach the different scenario sizes, we will look at the fraction of receivers reached by multicasts and the fraction of intended receivers among all receivers, which are shown in Figure 8.14. The y-axis in the right plot does not start at 0 for visibility.

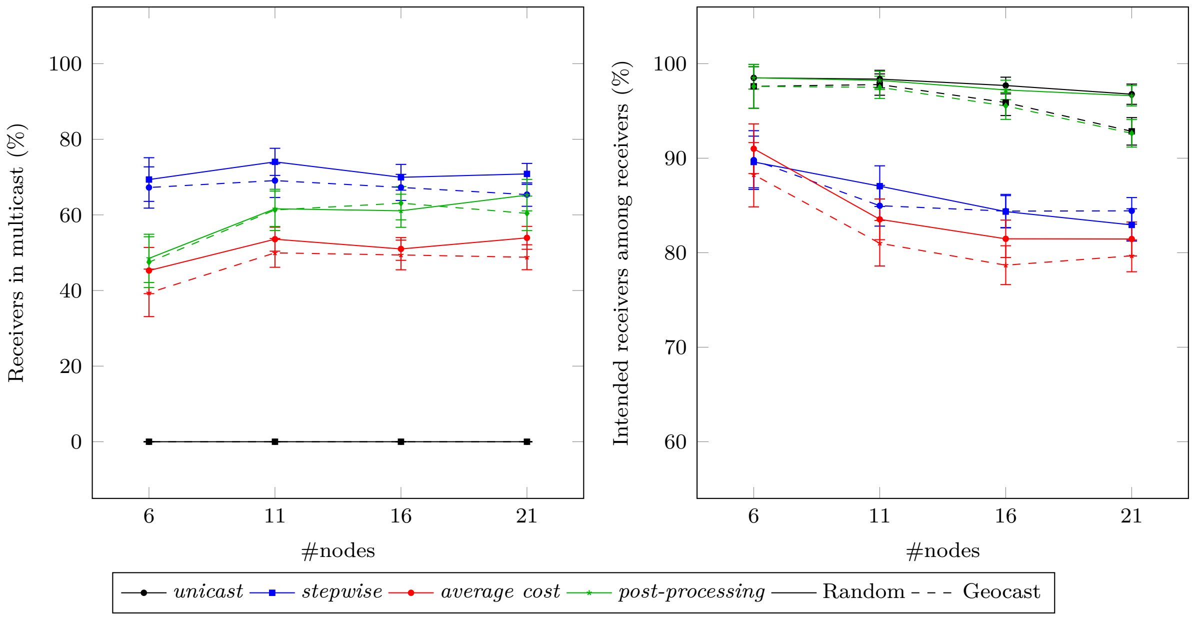


Figure 8.14: Average percentage of receivers addressed via a multicast and percentage of receivers that are intended receivers.

The mutual differences between the methods are about the same as before. The fraction of receivers reached via multicast for different number of nodes is roughly stable for *stepwise* and it only slightly increases for larger scenarios for *average cost* and *post-processing*. The fraction of intended receivers among all receivers, on the other hand, decreases for all methods, although it seems to stabilize for *stepwise* and *average cost*. In general both fractions are slightly smaller for geocasts.
So, for larger scenarios *stepwise* performs worse while the fraction of multicast receivers is stable. It thus seems that loosely choosing multicasts, while not checking if it is likely to benefit the result, works only up until a certain scenario size. Especially the method *average cost* proves its functioning in larger scenarios. For *post-processing*, this effect is less significant. In the case of a geocast, it addresses relatively less intended receivers for larger scenarios, leading to no further improvement for larger scenarios.

In conclusion, the method *stepwise* is inconsistent, while its performance is good in some scenarios. Especially for a larger number of intended receivers or nodes, the method *average cost* performs well. In these cases, the method *post-processing* performs only slightly less, whereas it is better in case of a small number of intended receivers or nodes.

Computation time

Next, we will evaluate the unicast and multicast methods for generating the transmission tree using Algorithm 3 in terms of their computation times. All heuristics are implemented in Python and run a laptop with an Intel i7-6700HQ processor with 16 GB RAM. For 1,000 snapshots of two scenario sizes, we generate the reduced graph, as well as a graph including all links, which is exhaustively determined (the two options shown at the top in Figure 5.8).

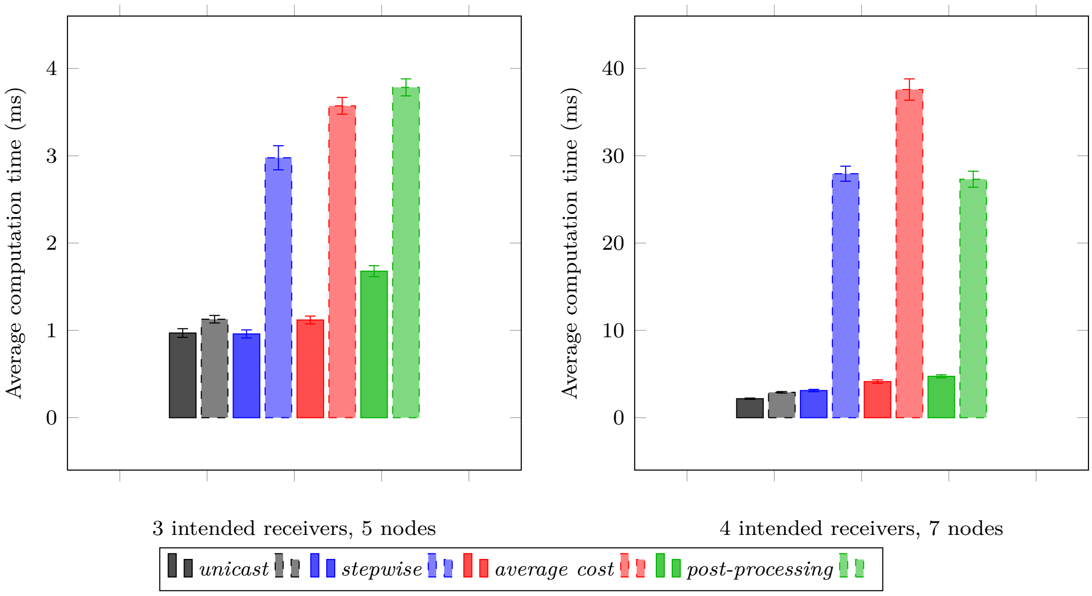


Figure 8.15: Average computation time of the heuristic methods, either using the reduced graph (filled) or all links (faded), for the smaller (left) and larger scenario (right).

The average computation time per snapshot and its 95% confidence interval for creating the transmission tree are given in Figure 8.15.
It can be seen that using all links as input, the difference between *unicast* and the heuristics including multicast selection is significantly larger, especially for the larger scenario (mind the y-axis scale). This shows that the reduction in multicast links due to the reduced graph leads to a much lower computation time, and is needed especially in larger scenarios.
The method *stepwise* uses the least computation time among the multicast heuristics, which is due to the fact that it includes many receivers in multicasts, thereby quickly covering nodes and excluding other links that have these nodes in common. Using the reduced graph as input, *post-processing* takes the most computation time. Although it has to compare newly created schedules with the unicast alternative, the computation time as compared to *average cost* is not so high, especially in the larger scenario. This can be explained considering that fewer multicast links have to be evaluated, since only those that can replace unicast links are valid. In the larger scenario with all links as input, it even has the least computation time.

In conclusion, we presented three different methods for selecting multicasts as an extension to the *DCST* algorithm. As compared to using only unicast links in this heuristic, the performance can certainly be improved by introducing the use of multicast links. This is especially the case for the methods *average cost* and *post-processing*. Both show better performance for an increasing number of intended receivers, as well as for an increasing number of nodes. However, the performance differs per scenario and underlying statistics show the differences in approach, from which can be concluded that the benefit of multicasts might not be fully exploited. To examine to what extent performance is lost, the heuristic methods will be compared with the optimal solution in the next chapter.

Chapter 9

Performance evaluation

The complete heuristic algorithm for the scheduling of transmissions needed to deliver one mmWave message to a set of intended receivers has now been fully defined. To answer research question 1, we are interested in the performance of the system when next to unicast links also multicast links are allowed. At first, it is interesting to see the difference in the optimal solution for both cases. Furthermore, we want to gain insight in the performance of the heuristic algorithm compared to the optimal solution. Afterwards, we will investigate the influence of multicasts on scenarios with a different number of highway lanes. Due to the high computation time of the mathematical optimization solver for large scenarios, only small scenarios could be evaluated with a sufficient amount of different snapshots.

Benefit and performance of multicast
The first scenario we consider uses 100 snapshots with 20 s separation from the five-lane US Highway 101 in [34] with 5 nodes. We assign the northernmost node as origin node and from the remaining nodes, three are randomly assigned to be intended receivers. The timeout was set to 100 ms. Each scenario was applied to both the mathematical optimizer, referred to as *optimal*, as well as to the different heuristic methods *stepwise*, *average cost* and *post-processing*. In order to answer research question 1, we also investigate the performance of the optimal solution when only unicasts may be used, by allowing only unicast links while pre-processing the scenario. We refer to this method as *optimal unicast*. Furthermore, for each method, we will use either the reduced graph as input or all links, which are determined exhaustively. Thus, all four main options as presented in Figure 5.8 will be evaluated.
We will compare all methods to *optimal* with all links as input. As metrics we will use the relative number of intended receivers reached before the timeout and the relative maximum delay in case the number of intended receiver reached is the same. Using the method *unicast* with reduced graph, in 81% of the scenarios all intended receivers were reached in time. The average values for both metrics including 95% confidence intervals are shown in Figure 9.1.
First, looking at *optimal unicast*, it can be seen on the left in Figure 9.1 that in terms of intended receivers reached the difference as compared to *optimal* is limited for this rather small scenario, both using the reduced graph and all links (remember that *optimal* with all links is the baseline). The performance difference is reflected in the number of intended

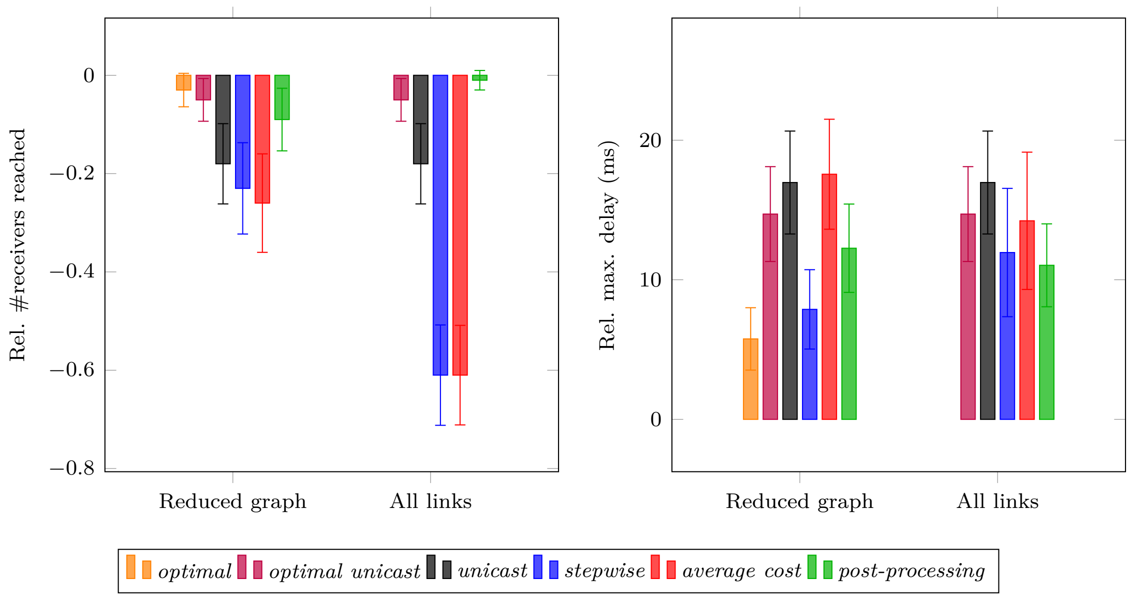


Figure 9.1: Average relative number of intended receivers reached before the timeout and relative maximum delay if the number of intended receivers is the same as compared to optimal, either using the reduced graph or all links.

receivers reached only when the created scheme is significantly worse. Therefore, it can be concluded that this is not often the case. On the other hand, as shown on the right in Figure 9.1, the delay using only unicasts is on average 15 ms higher (both using the reduced graph and all links), which is an increase of 20%. This shows that when used correctly, enabling multicasting is beneficial, even in a rather small scenario.

We will now look at the influence of using a reduced graph. For *optimal*, using the reduced graph results in a minor performance degradation as compared to using all links. In terms of delay it is on average 5.8 ms higher, which is an increase of 7.8%. Moreover, for *optimal unicast* and the heuristic method *unicast*, there is even no difference in performance between the reduced graph and all links. This shows that the graph reduction step performs well for unicast links only.

Now let us analyze how multicast schemes perform in comparison to *unicast*. Except for *average cost* using the reduced graph, the heuristic multicast methods perform better than *unicast* in terms of delay, but the difference is limited for this small scenario. The method *stepwise* performs best in case the reduced graph is used. However, in terms of intended receivers reached, *post-processing* performs better than *stepwise* and *average cost*, as by design *post-processing* will never perform worse than *unicast*.

Furthermore, using all links does not always result into performance improvement for the methods *stepwise* and *average cost*. This is due to the fact that if more multicast links are available, it is more likely that these methods choose to use them, while it is not guaranteed that those eventually lead to performance improvement. Using all links does have a positive effect on the method *post-processing*. It comes very close to the optimal solution in terms of number of intended receivers reached using all links. In terms of delay, an increase of 17% and 15% is observed, using the reduced graph and all links as input, respectively. This

Table 9.1: Metrics for scenarios with a different number of highway lanes and nodes.

#Lanes	**#Nodes**	**Vehicles per km per lane**	**Max. delay (ms)**	**95% CI (ms)**	**Fraction multicast (%)**
1	5	199.8	110.6	5.7	9.3
	7	181.8	160.7	12.2	3.5
3	5	136.4	81.2	9.6	57.3
	7	144.1	95.2	9.2	55.9
5	5	107.5	86.8	9.2	43.8
	7	111.2	102.8	15.0	56.9

result shows that there is still a significant gap between the heuristic method and the optimal solution.

Performance for different number of highway lanes

Using wider beams, multiple receivers can be reached at once using a high data rate, if these receivers are located at limited vertical distance from the transmitter. It seems therefore that multicasts are more likely to be used in scenarios with multiple highway lanes, in which vehicles are more horizontally spread. We will investigate whether this observation is true and what it means for the performance of the system. For this, we will use two scenario sizes; one with five nodes and three randomly assigned intended receivers and one with seven nodes and four intended receivers. Snapshots are taken from the dataset of [34] with 20 s separation. We evaluate 100 snapshots in which either vehicles of the first lane, the first three lanes or all five are included. We assign the northernmost node as assigned the origin node and select the intended receivers randomly from the remaining nodes. The timeout is set arbitrarily high (10 s), such that we can investigate the delay of the last intended receiver that was reached for all methods. We will provide the reduced graph as input to the mathematical solver (see Figure 5.8 for the procedure), in order to limit its computation time. For a fair comparison the heuristic methods also use the reduced graph.

As the traffic varies per lane, the average vehicular density per lane differs. For each scenario, we provide this value in the third row in Table 9.1. Also, the different scenarios lead to deviating average values for the maximum delay. The average maximum delay obtained by the optimal solution and its 95% confidence interval (CI) are shown in the fourth and fifth row, respectively. Furthermore, the rightmost row shows the fraction of receivers that are reached via a multicast in the optimal solution.

Concerning the latter, it can already be seen that indeed multicasts are less frequently used in a scenario with only one lane. For three lanes the fraction is even higher than for five lanes in the smaller scenario and it is similar in the larger scenario, which can be caused by the higher vehicular density per lane. Also, the limited width of the antenna beams might be a reason for this.

What the more frequent use of multicasts means for the performance of the system is pre-

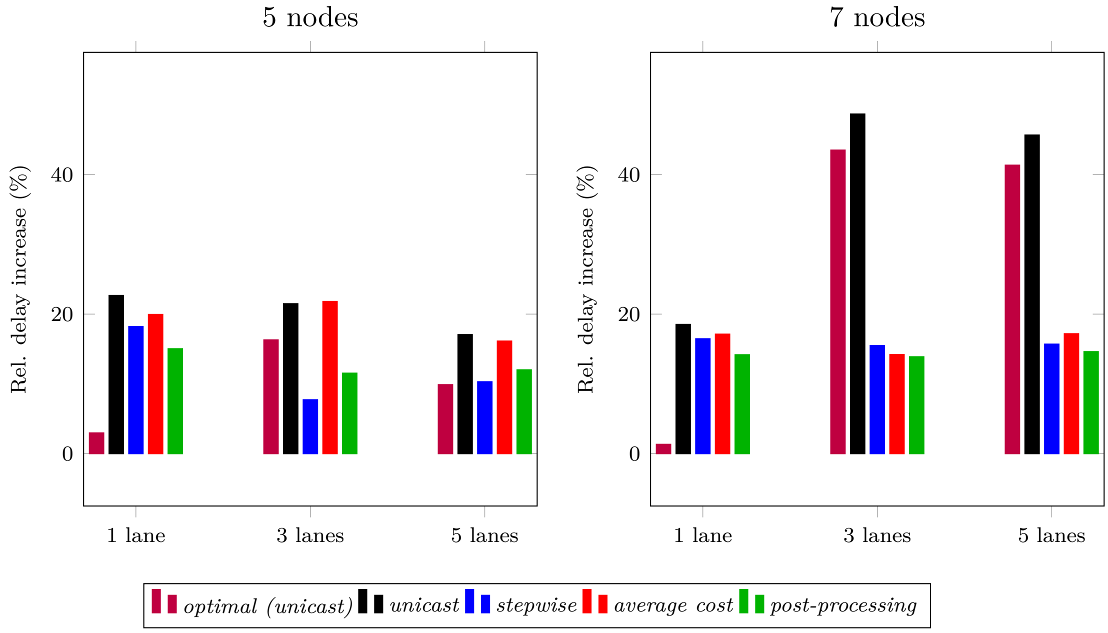


Figure 9.2: Average relative delay increase as compared to optimal using the reduced graph for snapshots with 1, 3 or 5 lanes and two scenario sizes.

sented next. We will use the relative increase in delay as compared to the optimal solution (with reduced graph as input) as metric. This value is determined for the average maximum delay of each method. The results for the smaller scenario are shown on the left in Figure 9.2, the results for the larger scenario are presented on the right.

We will first look at the optimal using unicast links only. For one lane the delay increase as compared to the optimal solution including multicasts is marginal for both scenario sizes, namely 3.0% (left) and 1.3% (right). On the other hand, for three and five lanes, the increase for the optimal solution is significantly larger, namely 16.3% and 9.9% in the smaller scenario on the left and even 43.5% and 41.3% for the larger scenario on the right. Thus, the number of highway lanes affects the performance of the system for the optimal solution, which follows from the multicast usage. The effect is strongest for three lanes, which is also in line with the fraction of multicasts used.

Next, we will look at the heuristic methods. For three and five lanes, the difference between *unicast* and the multicast heuristics is also significantly larger as compared to one lane. This is especially the case in the larger scenario on the right. In the smaller scenario, this is not so clear for *average cost*, which was observed earlier when the reduced graph was used as input. Apart from this, the results between the heuristic multicast methods do not vary significantly for the different number of lanes and are similar to what has been shown before.

In conclusion, we have shown that the use of multicasts is indeed beneficial when looking at the optimal result for routing and scheduling mmWave geocasts. This is especially the case for highway scenarios with multiple lanes. Furthermore, if used correctly, multicasting can improve the result of a heuristic algorithm. Also here, the performance improvement is more significant for scenarios with multiple lanes.

Chapter 10

Distributed message scheduling

As mentioned before, the directional constrained Steiner tree heuristic is designed to let the origin node calculate the complete transmission tree. However, when any node may generate a mmWave message at any time, the spectrum should be shared in a sophisticated way. If the schedule is already congested by one message, a message generated later might not be handled before the timeout, whereas it could if the previous message would be transmitted via a different path. Thus, a dynamic scheduling procedure is needed.

Furthermore, we will provide a procedure to announce mmWave transmissions, which is necessary such that the receiver can point its antenna towards the direction of the transmitter. In [24], the scheduling is regulated using a static RSU. However, in case no RSU is available or its coverage is insufficient, a distributed algorithm is needed. Some additional time is taken into account to distribute this information via omnidirectional antennas for 4G communications in [23]. In [25], position information and the scheduling decision is assumed to be known to all nodes, however they propose to use the sub-6GHz interface for real-life implementation. A schedule using asynchronous sub-6GHz beacons is presented in [2], but this system takes into account neither multicast nor relaying.

The system considered here follows the latter approach by adapting the directional constrained Steiner tree heuristic as presented in Chapter 8. First we will explain what this means for the design and which assumptions are made. Then, the procedure to execute the heuristic algorithm as described in the previous chapters in a distributed way via sub-6GHz beacons is presented. Finally, we evaluate this procedure by applying it to scenarios in which multiple messages are generated.

10.1 Sub-6GHz control

Apart from a mmWave radio interface, the nodes are also equipped with a sub-6GHz interface. For the control of mmWave transmissions, the nodes make use of CAMs via the sub-6GHz band, as specified in the ETSI standard [8]. In order to make mmWave communication possible via these sub-6GHz beacons, only slight adjustments to this standard are needed, which are explained as follows.

Firstly, for a node to be able to receive a transmission, it should point its antenna towards the

transmitter and use the right decoding scheme. Therefore, before transmitting a mmWave frame, at least information about the transmitter, receivers, starttime and MCS to be used should have been transmitted in a CAM. A regular CAM already includes useful data, such as the reference position, heading direction, speed, acceleration and dimension of the vehicle. From this, the positions of nodes at the time of transmission and the number of obstacles obstructing a link can be estimated.
Following, an additional condition on the generation of a CAM is necessary. The upper and lower transmission interval of CAMs are 100 ms and 1000 ms, respectively. This is ensured by checking whether a new CAM needs to be generated at least every 100 ms. A regular CAM needs to be generated if the transmission interval between CAMs will otherwise exceed 1000 ms or if certain thresholds related to the dynamics of the node are exceeded. The necessary additional condition can be formulated as follows:

Condition. *New information for the control of mmWave transmissions is available.*

This ensures that if new control information is available, a CAM is generated at most 100 ms later. Control information may be an announcement of mmWave transmission, or a newly generated mmWave message. Nodes generally check whether they should send a beacon at different points in time, i.e. the beacons are not synchronized.

Additionally, a distributed and real-time system leads to some limitations for the routing and scheduling algorithm. Control information should be available at the time when it needs to be sent. In the ETSI standard this is, amongst others, accounted for by specifying a hard requirement on the generation time of a CAM. This is defined as the time between the generation trigger and the time at which the CAM is delivered to the networking transport layer, and should be less than 50 ms. Moreover, each CAM has a time stamp, which should correspond to the time at which the reference position of the node was determined. This means that a node has at most 50 ms to generate the control information it wants to transmit in a CAM, without possibly having to force itself to send a less recent reference position. An additional incentive to keep the computation time short is that it allows to use more recent information, as new beacons may arrive at all times. The heuristic algorithm as presented in the previous chapters was designed with low complexity and deterministic behavior in mind, making it suitable for distributed computing.
In reality, the distribution of a CAM itself does introduce some delay and it is not guaranteed that CAMs arrive. Without changing the essence of the problem, we assume here that all nodes in the system are in the sub-6GHz range and will receive the CAMs without any delay. Therefore, each node can generate the reduced graph for a topology, which it derived from the information included in sub-6GHz beacons. Thus, in terms of knowledge about position and scheduling, the distributed algorithm is similar to the optimal situation.
As explained, the authors of [2] proposed to make use of an RTS sent by the transmitter. The receiver that reacts first with a CTS plans the transmission for itself by announcing the start time. However, in order to allow relaying, we needed to create specific routes to reach the destinations. Making use of the global scheduling knowledge, we will allow any node to

generate the routes and mmWave transmission schedule, which it can directly announce in a RTS-like beacon. It is assumed that every node overheard the RTS, such that the schedule information is distributed and no CTS is needed.
Nevertheless, a significant limitation as compared to the optimal situation is that the notion of a message, together with its origin node, generation time and intended receivers, are only known to the origin node once that message is generated, and to other nodes once they receive a beacon with this information. This means that the complete schedule cannot be determined upfront, such that the method *post-processing* is not suited anymore. We will therefore modify the *DCST* heuristic only for the methods *stepwise* and *average cost*. The procedure we propose to make distributed scheduling possible, while taking the restricted update frequency by offloading the control to sub-6GHz beacons into account, is described hereafter.

10.2 Distributed procedure

Instead of fixing the transmission schedule at the generation time of a message, we need to allow dynamic scheduling of several concurrent messages. Therefore, we will generate the transmission tree in steps, which are chosen to take place on sub-6GHz beacon intervals, as these are the points in time where transmissions will be announced. The procedure of generating the transmission tree considering multiple active messages is described next.
Every node should react on two events, namely upon the reception of a beacon from another node and at its *beacon opportunity*, which is the time at which it should check whether it should send a beacon.
First, we will explain the procedure for a node when receiving a beacon. If the beacon includes a message announcement, it executes the following three steps (regardless of whether it is an intended receiver for the message):

1. Include the new message in the list of active messages $\mathcal{M}$;
2. Generate the reduced graph via Algorithm 2 for this message;
3. Execute lines 1-10 of Algorithm 3, to initialize the generation of the transmission tree for this message.

At this moment, each node that received the announcement can generate a part of the transmission schedule for this message. How it determines this will be explained later.
A node can also receive a beacon containing a transmission announcement. Such an announcement includes which message is contained, the transmitter, the receiver set, start time, and MCS to be used for the transmission. Every node that receives such an announcement executes the following two steps:

1. Execute lines 25-29 of Algorithm 3, in order to;
    - Update the set of visited nodes $\mathcal{V}$ for this message;
    - Update the set of nodes still to find $\mathcal{U}$ for this message;

- Update the transmission tree that is set up so far;
- Update the delays of nodes for the transmission tree that is set up so far.

2. Update the individual schedule for transmissions to transmit or receive.

The first step is needed to participate in the procedure of distributed transmission tree generation. By administering an individual schedule as in the second step, a node knows when to transmit or receive a mmWave transmission.
Let us now explain when a node should send a beacon with control information. Just before a nodes' beacon opportunity, it will follow the process described next and schematically presented in Figure 10.1.

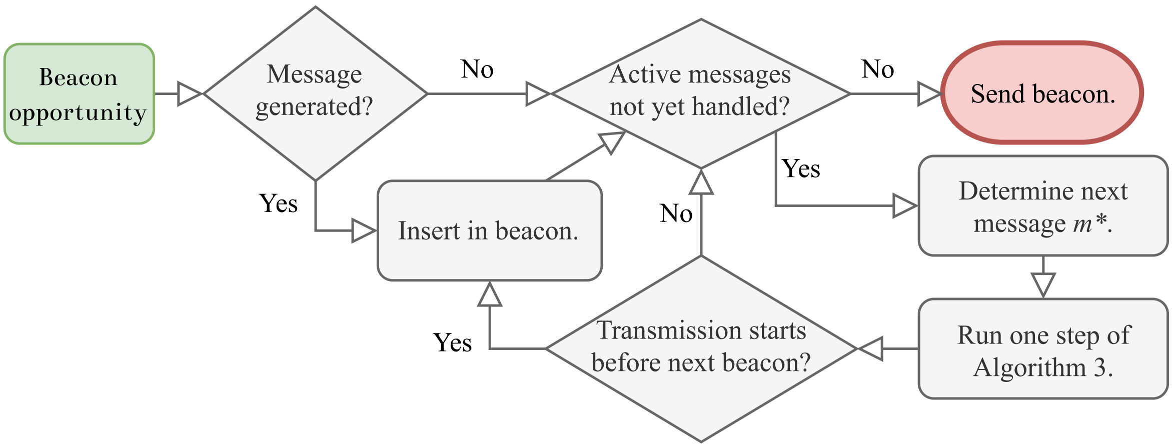


Figure 10.1: Procedure of announcing transmissions in beacons with multiple concurrent messages.

First, if the node itself generated a message that it did not yet announce, it will insert the announcement in the beacon to inform other nodes about the message, its intended receivers and timeout.
Furthermore, the node will determine for which message a new path in the transmission tree is calculated. It might have received several message announcements, such that multiple active messages are waiting in a queue. A rather trivial approach, namely *First In, First Out (FIFO)* can be applied here. Using this procedure, transmissions for the message m^* that was generated first will also be scheduled first. This is done until all intended receivers are served or the timeout is violated.
We introduce another metric here, which we will call *Residual Time Per Receiver (RTPR)*. Using this metric, for each message m_j the residual time before the timeout divided by the number of intended receivers not yet scheduled is calculated:

$$RTPR_{m_j} = \frac{t_{m_j}^{out} - t_b}{|\mathcal{U}|}, \tag{10.1}$$

where t_b is the time at which the beacon announcing the next transmission will be sent.
The next path to add in the tree should be for the message m^* for which this value is minimum:

$$m^* = \min_{m_j \in \mathcal{M}} RTPR_{m_j}. \tag{10.2}$$

After determining m^*, the node will run one step of Algorithm 3 for this specific message. One step meaning that the next path will be determined via the procedure of lines 12-29. A slight modification of the algorithm is needed to schedule multiple messages. Namely, a transmission cannot start directly after the last transmission in which the transmitter participated, since it should take already scheduled transmissions for other messages into account. Therefore, lines 16-17 are replaced by the function `insertTransmission()`. This function calculates the first possible starttime such that the transmission does not overlap with transmissions in which the transmitter or any of the receivers participates.

If the node concludes from this function that a transmission for m^* should start before its subsequent beacon opportunity (meaning the one after the current for which the content is determined here), it will announce this transmission, whether or not it participates in it. By announcing only when the start time of the transmission for m^* is before its subsequent beacon, we postpone the scheduling decision as far as possible, while still assuring that the transmission is announced before it is planned to start. This works even if at some beacon opportunities no beacon is sent (and thus the beacon interval is aperiodic), as the procedure is executed at least every 100 ms.

A node might announce multiple transmissions in one beacon. Therefore, it repeatedly determines whether there exist active messages for which transmissions can be scheduled. Only once for none of the active messages a transmission should be scheduled that starts before the subsequent beacon opportunity of the node that runs the procedure, the algorithm stops and the beacon is sent, if any content for it is created.

In Figure 10.2, it can be seen how the schedule for mmWave transmissions of the input scenario of Figure 3.1 is announced via beacons. The green boxes show the node that announces a specific message and/or frame. The beacon opportunities for the different nodes are shown by the colored markers underneath the timeline.

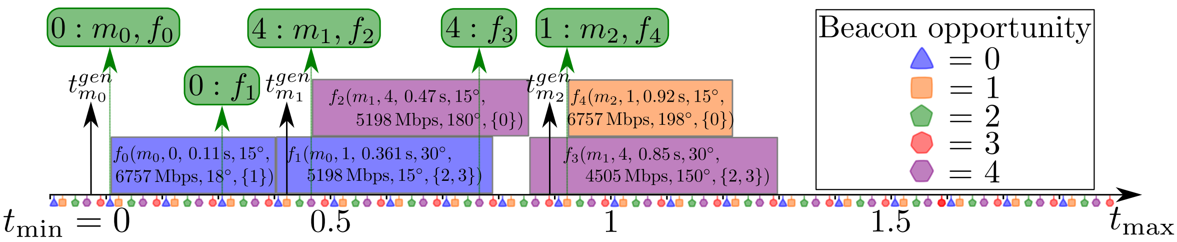


Figure 10.2: Schedule of transmissions for the input of Figure 3.1 announced in beacons.

It can be seen that the frames f_0, f_2 and f_4 are slightly delayed because of the time difference between message generation and the next beacon opportunity of the origin node. Beacon opportunities of other nodes may exist during this period, but because they are not yet aware of the message, they cannot announce frames for it. Subsequent frames containing the same message (e.g. f_1 and f_3) are not further delayed, as they can be planned in advance and announced by any node in its last beacon opportunity before the frame starts. For example, f_1 will be sent by node 1, but it is announced by node 0, as the frame starts just before node 0's beacon opportunity.

Shortcomings

Several shortcomings of the presented method can be identified. First of all, the shortest path calculation as presented in lines 1-4 of Algorithm 3 does not take already scheduled messages into account. Therefore, while generating the transmission tree, if a shortest path includes a node that is busy sending another message, there might have been a different path resulting in a lower delay. However, calculating the shortest path at each step in Algorithm 3 by taking the schedule into account leads to additional computational overhead. On the other hand, the algorithm is still free to choose another transmitter or intended receiver which does not include a busy node on the shortest path. Therefore, additional delay due to busy nodes can still be mitigated if other visited nodes or nodes still to be found are available.

Secondly, the metric for determining the message to handle does not include the delay of the to-be-scheduled transmission, as it is hard to predict upfront. As Algorithm 3 decides upon a transmission after it considers every possible transmitter and intended receiver combination, it could also do this for all active messages. However, this is again a trade-off between performance and computational complexity. In this case, using the more elaborate way only has an influence if the transmission delay between different messages varies a lot.

10.2.1 Evaluation

In this section, we will investigate the performance of the distributed procedure by a simulation. The algorithms are executed on one machine, however the procedure is followed as if it was executed on different nodes. That is, at each beacon opportunity of any node, part of the transmission tree will be calculated for the message m^*. Each node has all the information it needs, since we assumed perfect delivery of CAMs. We will evaluate the use of the queuing metric *RTPR* as compared to *FIFO*. It is also interesting to see how the multicast methods compare to *unicast* in case of multiple messages. We will consider two scenarios in which two messages m_0 and m_1 are generated. Although only two messages are examined, the results can be generalized to more, which we will explain later.

The two scenarios are identical, except for the generation time of the second message. In the *concurrent* scenario, the difference in generation time is only 50 ms, whereas in the *separate* scenario, m_1 is generated after the timeout of the first. We use 11 nodes and 6 intended receivers per message, which are located in a geocast area that is the same for both messages. However, the messages have a different origin node; for m_1 it is the node farthest from the geocast area, for m_2 it is the one but farthest node. We will look at the average number of intended receivers reached per message for 100 snapshots from [34] with 20 s separation. The timeout was set to 200 ms for both messages, thus in the separate scenario, m_1 was also generated 200 ms later. The results are shown in Figure 10.3 for both the *RTPR* and *FIFO* queuing strategies.

As expected, in the separate scenario, more receivers can be reached within the timeout. In the concurrent scenario, the effect of fixing a schedule for one message becomes clear, because for m_1 significantly less intended receivers could be reached. The schedule is congested by transmissions for m_0, which are often conveyed by nodes that are also needed for distributing

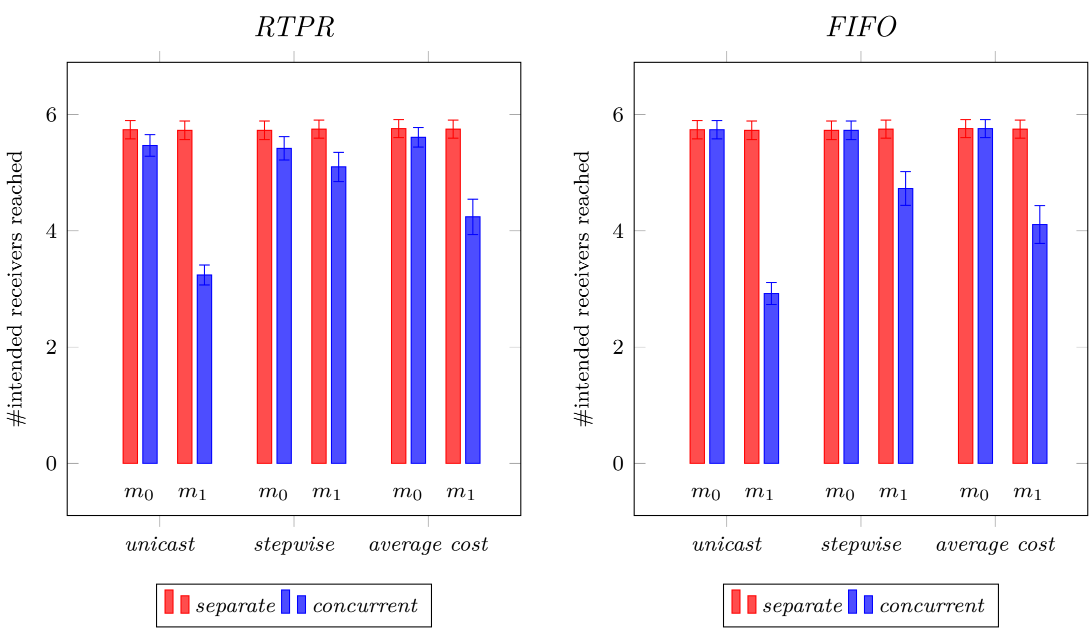


Figure 10.3: Average number of intended receivers reached before the timeout for two messages generated concurrently or separately.

m_1, since the geocast area is shared. This burdens the delivery of m_1 and the same would hold for other more messages generated shortly after m_1. In this case, additional delay would be introduced, depending on the separation in message generation times and node usage.

Using the *FIFO* strategy, m_0 was delivered to exactly the same number of intended receivers in both scenarios, which results from its priority in the schedule. On the other hand, the number of receivers reached for m_1 in the concurrent case is on average lower than using *RTPR*. Therefore, although the difference in terms of total intended receivers reached is not significant, the method *FIFO* is less fair.

Furthermore, it follows from the figure that there is only a minor difference between the methods *unicast*, *stepwise* and *average cost* for the separate scenario - it is even barely visible. On the other hand, both multicast methods perform significantly better for m_1 in the concurrent scenario. It must be said that in the separate case it is less difficult to reach all intended receivers, which causes less variation in the result. However, it shows additional benefit of using multicasts in scenarios with messages generated closely after each other. What is interesting to see is that the method *stepwise*, which uses multicasts the most often, performs best. This strengthens the view that multicasts pay off in congested schedules. This also follows from the reasoning that the additional delay introduced by a multicast is relatively small as compared to when a transmission is delayed due to other concurrent messages. If then via a multicast multiple receivers are served at once, fewer nodes will later form a bottleneck for the other messages.

Next, we also evaluate the performance of *stepwise* and *average cost* as compared to *unicast* in terms of delay for both the concurrent and separate scenario. To eliminate differences in the number of intended receivers reached, the timeout was set to 10 s. In the separate scenario m_1 is thus also generated 10 s later, whereas this is still 50 ms in the concurrent scenario. The average relative maximum delay with 95% confidence intervals is shown in Figure 10.4. The delay resulting from the difference in message generation time and next beacon opportunity of the origin node is incorporated here. Results using queuing metric *RTPR* are shown on the left, those using *FIFO* on the right.

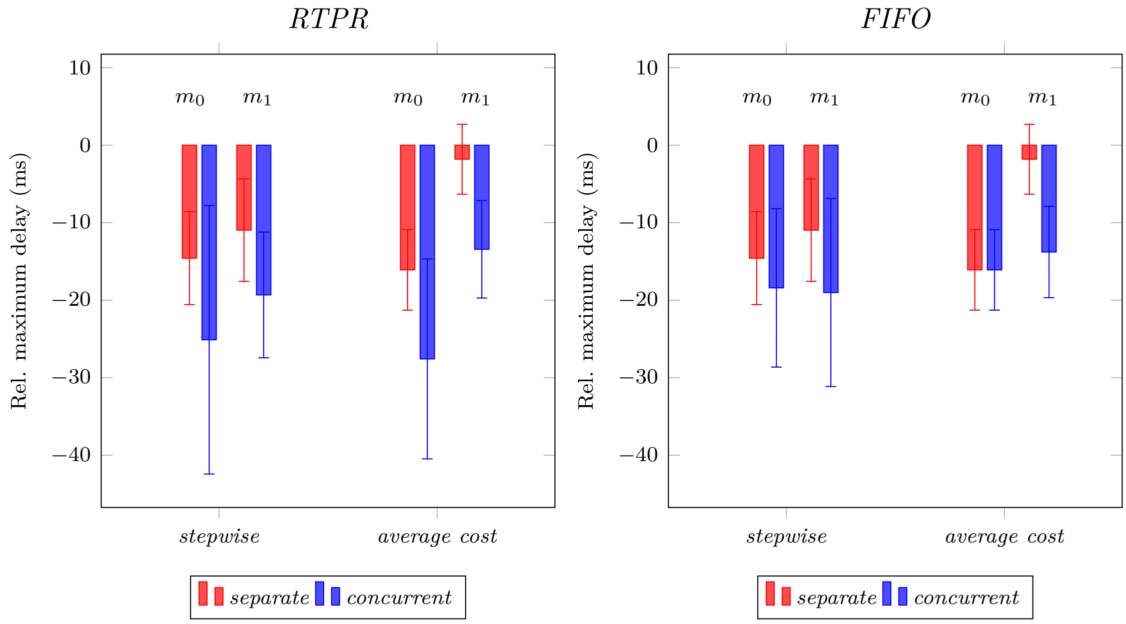


Figure 10.4: Relative maximum delay as compared to unicast for two messages generated concurrently or separately.

The relative difference between both multicast methods and *unicast* is smaller for m_1 in general, as its origin node is closer to the geocast area.
Furthermore, for both messages, a larger difference in delay can be observed in case of concurrently generated messages. This again shows that the multicast methods come out even better in congested scenarios. However, the confidence intervals are large, especially for the concurrent scenario, showing more variation in the result. This follows from the shortcomings we discussed and the sub-optimality of the multicast selection methods as shown in Chapter 8. The impact of the multicast methods seems stronger for the queuing method *RPTR*, especially for m_1. Although using *FIFO* the message that was generated first gets priority for scheduling, when no transmission could be scheduled before the next beacon, nodes are allowed to schedule transmissions for the other message. Hence, there is still a difference in result between the concurrent and separate scenario for m_0, but it is limited.

Summarizing, we have defined the system needed for distributed control of mmWave transmissions via sub-6GHz beacons. Next, a procedure was proposed such that the *DCST* heuristic can be executed in a distributed manner by separating it in several steps. Newly generated messages and scheduling decisions made should be distributed via the sub-6GHz layer to provide the relevant information to let any node calculate part of the schedule. Finally, we showed that using multicasts is even more beneficial in case of congested mmWave schedules due to multiple concurrent messages.

Chapter 11

Conclusions and future work

With the increasing amount of data generated by sensors, geocasting in vehicular networks requires higher data rate links, which can be achieved on mmWave frequencies. However, due to the high propagation loss imposed by these frequencies, beamforming is needed to obtain sufficient range. The characteristics of such a system as compared to one with omnidirectional antennas asks for different approaches in routing and scheduling, especially when various beamwidths and data rates are considered. Wider beamwidths allow for reaching multiple receivers at the same time, at the cost of transmission range. Various data rates also influence the coverage, while relaying and spatial sharing might help to reach intended receivers in a short time. Furthermore, it has been shown in previous works that offloading control information to a separate sub-6GHz band promises to overcome the overhead that is induced when solely using the mmWave layer. To investigate these points, we presented three research questions in this thesis, which we will answer in summary hereafter.

Answering research questions

Following the previous work on mmWave routing and scheduling, we presented the following research questions:

1. *To what extent can mmWave geocasting benefit from using multicasts while considering multiple beamwidths and data rates, relaying and spatial sharing?*
2. *How can a mmWave geocast be efficiently routed and scheduled using a lightweight algorithm?*
3. *How can multiple mmWave geocasts be scheduled in a distributed manner using sub-6GHz beacons?*

Firstly, to answer these questions, we defined a system considering static nodes. To model the mmWave communication, we used a realistic propagation model, which takes obstacles into account. Furthermore, the antenna model used allows for multiple beamwidths, which leads to diverse antenna coverage for various data rates. Furthermore, nodes can only transmit or receive one mmWave transmission at a time. Interference between concurrent transmissions was not taken into account.

Then, we formulated a mixed-integer linear program and implemented it using the mathematical optimizer Gurobi to evaluate the potential performance of the system and eventually answer research question 1. The program uses two objectives with decreasing priority. The first objective is to maximize the amount of intended receivers that are reached before the timeout of the message. The second is to minimize the delay between message generation and delivery to the last intended receiver that is reached.
Next, we have shown that several steps are needed to develop a heuristic algorithm that considers the various elements of the system. Together, these steps form an answer to research question 2. The first step, *link assessment*, determines whether a link exists between a transmitter and a receiver set and which beamwidth and transmission direction angle should be used to maximize the data rate of the link. We have shown the complexity originating from using a realistic antenna model with various beamwidths and data rates. It is not trivial to find the optimal beam and point the antenna in the right direction, resulting in performance loss for the heuristic approach we proposed. About 16% of the links used by optimal routing and scheduling are not detected by the heuristic algorithm, whereas the computation time is only 1.5% that of an exhaustive search.
Following, we chose to take advantage of the farsighted vision obtained via information on sub-6GHz beacons, such that efficient routes up to the destination can be found. However, due to the exponential increase in the number of links for an increasing number of nodes, finding these routes easily leads to high computation times. Therefore, we propose to make use of the step *graph reduction*, in which a directed hypergraph is generated that only includes the nodes and links that are likely to contribute to the result. For this, we introduce the notion of MPRs, which are nodes that are likely to be used for relaying. Links between MPRs are included only if they comply to a condition related to their MPRDs, the intended receivers for which they are a relaying node. We have shown a reduction of 72% in the number of links, while 95% of the links selected by the optimal solution are included in the reduced graph.
Subsequently, with the reduced graph as input, we presented a heuristic method for *transmission tree generation*. Such a tree specifies which links need to be used, and how transmissions using these links should be scheduled. We developed an algorithm based on a modification of the constrained Steiner tree heuristic from [33] to make it suitable for directional antennas, which we called *DCST*. We proposed a dynamic cost that incorporates the delay of a transmission after insertion in the schedule. To control the order of transmissions, we introduced an additional factor, which enlarged the chance of reaching more intended receivers and reduced the maximum delay, as we showed by simulations. Furthermore, we showed that *DCST* outperforms more simple heuristic methods.
Next, we presented three distinct methods to include multicasts in the tree. The first, *stepwise*, loosely includes multicasts if during tree generation, the receiver of the unicast choice is covered in a *relevant* multicast. A relevant multicast is defined as a multicast that has a lower transmission duration than the minimum spanning tree formed by unicasts that cover all receivers. The method *average cost*, on the other hand, only uses a relevant multicast link when it has the lowest cost, but to let it compete with a unicast, the cost is divided by the number of receivers of the link. Lastly, *post-processing* first evaluates the tree using

unicast links and replaces these by multicasts only if this guarantees to reach at least the same amount of intended receivers, and at least one of the receivers is reached with lower delay.

We have shown that, although it performs well in some scenarios, the method *stepwise* is not consistent. This grounds our observation that assessing whether a multicast should be used is difficult when relaying is considered. As compared to the heuristic method with only unicast links, using the method *average cost* often results into more receivers reached before the timeout, where this is always the case for *post-processing*, as follows from its design. In terms of delay, both methods perform better as the number of intended receivers increases and for an increasing number of nodes with a fixed percentage of intended receivers. When all nodes are intended receivers in scenarios with 15 nodes, a delay reduction of 19.3% as compared to the heuristic method using unicast links only can be achieved for *average cost*, while for *post-processing* this is 14.6%. However, by looking at the underlying metrics that follow from the choices made by the methods, we can conclude that both methods do not fully exploit the use of multicasts.

Furthermore, for generating the transmission tree, we have shown that it pays off to use a reduced graph in terms of computation time, especially for larger scenarios. For the methods *stepwise* and *average cost*, it also leads to performance improvement, since the multicasts that are not likely to contribute to the result are filtered out and cannot be chosen.

To answer research question 1, we examined the performance of mmWave routing and scheduling using unicasts only and when multicasts were allowed, both in the optimal case and using the presented heuristic methods. Due to the high computation time for the mathematical solver, the optimal solution could only be obtained for relatively small scenarios. In these scenarios, the difference of the heuristic multicast methods as compared to using unicast links only is limited. Furthermore, the benefit of using multicasts in the optimal case is not so large in terms of intended receivers reached, but a clear reduction in the maximum delay can be seen. This is especially the case when considering a highway scenario with multiple lanes. In case only one lane is considered, multicasts will not be frequently used, which results into only limited improvement in these scenarios. Also using the heuristic methods, the benefit of multicasts becomes more evident when vehicles are distributed on multiple lanes.

Finally, to answer research question 3, we designed a procedure to transform *DCST* into a distributed algorithm. To make this possible, control information for scheduling of mmWave transmissions is added to beacons sent via the sub-6GHz layer. Specifically, a beacon contains the necessary information to receive a transmission and to update the transmission tree. This allows any node that received the beacon to calculate a new path in the tree, insert the tranmissions in the schedule and announce them in the next beacon again. The announcements are delayed as far as possible to keep the schedule flexible, such that newly generated messages can still be inserted. We presented the strategy *RTPR*, that allows transmissions for the message with the least residual time per intended receiver to be scheduled first. Compared to the method *FIFO*, in which transmissions for messages that are generated first are also scheduled first, *RTPR* is more fair. Furthermore, it follows from simulations that the heuristic multicast methods perform relatively better in case the schedule is congested by multiple concurrent messages. However, it can be seen that especially in terms of delay, it leads to varying results.

Suggestions for future work

From the work presented in this thesis, various aspects arise that we suggest to investigate further. First of all, we suggest some changes to the system, which would make it more realistic. We made the assumption that nodes are static for the duration of a scenario. However, one of the characteristics of vehicular networking that sets it apart from other wireless networks is the high mobility of nodes. In the distributed version we presented, the transmission tree is already separated into steps. Therefore, the reduced graph can be set up again at these moments using the latest position information. Since the information is only updated on beacon intervals, it is necessary to estimate the position at the time of transmission. This is possible by using the data sent in beacons, which includes the speed, acceleration and heading of the vehicle. However, mobility might cause the reduced graph to change significantly over time, possibly making routes calculated in earlier steps not valid anymore. Sophisticated methods are needed to recover from these situations.

Moreover, we suggest to look at the influence of interference between transmissions. To exploit spatial sharing, unicast transmissions are now frequently scheduled concurrently by our heuristic algorithm. However, it is expected that this is not always possible, although using beamforming the effect is limited. If concurrent unicasts interfere with each other, it is expected that replacing these with a multicast transmission would be an appropriate alternative. On the other hand, wider beams might lead to more interference in general, which makes it an interesting trade-off to investigate. Eventually, the use of an algorithm that can predict interference and adapt the routing and scheduling accordingly is likely to improve the performance in realistic scenarios.

Other propagation characteristics that influence the channel quality, such as reflection, are also interesting to examine. This introduces additional constraints to the link assessment step, in which already performance degradation was observed. Other methods to assess the dynamic link quality that perform at least equal to traditional beam training techniques are likely needed. Channel state information from the sub-6GHz layer might contribute to this, as presented in [19] and [20].

As the graph reduction step significantly reduces the computation time and even leads to better performance for some of the heuristic methods, it might pay off to improve it even further. Still, many multicasts with distinct receiver sets are included in the graph, but only few prove to be beneficial. Smart grouping of receivers based on the mutual path loss between MPRs can support the selection of useful multicasts and might reduce the graph even further. Other properties of the graph that can exclude links early on are interesting to investigate as well. A better pre-processed scenario also leads to a reduction in computation time for the mathematical solver, such that the optimal solution can be obtained for larger scenarios.

Next, we have shown that the heuristic multicast selection method *stepwise* is not consistent in its performance, while *average cost* and *post-processing* do not fully exploit the benefit of multicasts and perform still worse than the optimal solution. A more sophisticated multicast selection procedure would therefore be required to improve the algorithm as a whole, however it should have limited computational complexity. A method that compares the resulting schedules, like *post-processing* does, seems to work well, but it should not be restricted to follow the unicast tree.

Lastly, we used several assumptions concerning sub-6GHz beacons for the distributed algorithm. For a realistic system, the procedure should be made robust against significant delay or lack of delivery of beacons to mitigate mismatches between scheduling decisions. Furthermore, we have addressed some shortcomings of the algorithm that concern the scheduling of multiple messages. Especially including multicasts is an open research direction, as it proves to perform well in congested scenarios, but it has varying results.

Bibliography

[1] C. Campolo, A. Molinaro, and R. Scopigno, "From today's VANETs to tomorrow's planning and the bets for the day after", *Vehicular Communications*, vol. 2, no. 3, pp. 158–171, 2015, ISSN: 2214-2096. DOI: `https://doi.org/10.1016/j.vehcom.2015.06.002`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S2214209615000418`.

[2] B. Coll-Perales, J. Gozalvez, and M. Gruteser, "Sub-6GHz Assisted MAC for Millimeter Wave Vehicular Communications", *IEEE Communications Magazine*, vol. 57, no. 3, pp. 125–131, 2019. DOI: `10.1109/MCOM.2019.1800509`.

[3] B. Coll-Perales, M. Gruteser, and J. Gozalvez, "Evaluation of IEEE 802.11ad for mmWave V2V communications", in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2018, pp. 290–295. DOI: `10.1109/WCNCW.2018.8369031`.

[4] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Beam alignment for millimetre wave links with motion prediction of Autonomous Vehicles", in *Antennas, Propagation RF Technology for Transport and Autonomous Platforms 2017*, 2017, pp. 1–8. DOI: `10.1049/ic.2017.0023`.

[5] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States", *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011. DOI: `10.1109/JPROC.2011.2132790`.

[6] T. ETSI, "Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band", *EN 302 663 V1. 2.1, July 2013*,

[7] 3GPP, *Summary of Rel-14 Work Items (Release 14)*, TR 21.914, May 2018.

[8] EN 302 637-2 V1.4.1, "Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service", ETSI, Standard, Apr. 2019.

[9] EN 302 637-3 V1.3.0, "Basic Set of Applications; Part 3: Specifications of Decentralize Environmental Notification Basic Service", ETSI, Standard, Aug. 2018.

[10] C. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding Geocast: Time–Stable Geocast for Ad Hoc Networks", in *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '05, Cologne, Germany: Association for Computing Machinery, 2005, pp. 20–29, ISBN: 1595931414. DOI: `10.1145/1080754.1080758`.

[11] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune, "Geographic routing protocols for Vehicular Ad hoc NETworks (VANETs): A survey", *Vehicular Communications*, vol. 11, pp. 20–31, 2018, ISSN: 2214-2096. DOI: `https://doi.org/10.1016/j.vehcom.2018.01.006`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2214209616300183`.

[12] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00, Boston, Massachusetts, USA: Association for Computing Machinery, 2000, pp. 243–254, ISBN: 1581131976. DOI: `10.1145/345910.345953`.

[13] J. Tian, L. Han, and K. Rothermel, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks", in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 2, 2003, 1546–1551 vol.2. DOI: `10.1109/ITSC.2003.1252743`.

[14] J. Choi, N. González Prelcic, R. C. Daniels, C. R. Bhat, and R. W. H. Jr., "Millimeter Wave Vehicular Communication to Support Massive Automotive Sensing", *CoRR*, vol. abs/1602.06456, 2016. arXiv: `1602.06456`. [Online]. Available: `http://arxiv.org/abs/1602.06456`.

[15] Y. Ghasempour, C. Silva, C. Cordeiro, and E. Knightly, "IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi", *IEEE Communications Magazine*, vol. PP, pp. 1–7, Oct. 2017. DOI: `10.1109/MCOM.2017.1700393`.

[16] T. Zugno, M. Drago, M. Giordani, M. Polese, and M. Zorzi, "Toward Standardization of Millimeter-Wave Vehicle-to-Vehicle Networks: Open Challenges and Performance Evaluation", *IEEE Communications Magazine*, vol. 58, no. 9, pp. 79–85, 2020. DOI: `10.1109/MCOM.001.2000041`.

[17] 3GPP, *Study on NR Vehicle-to-Everything (V2X)*, TR 38.885, Mar. 2019.

[18] 5GCAR, *Final 5G V2X Radio Design*, Deliverable D3.3, May 2019.

[19] C. Anjinappa and I. Guvenc, "Angular and Temporal Correlation of V2X Channels Across Sub-6 GHz and mmWave Bands", May 2018.

[20] Z. Ali, A. Duel-Hallen, and H. Hallen, "Early Warning of mmWave Signal Blockage and AoA Transition Using sub-6 GHz Observations", *IEEE Communications Letters*, vol. 24, no. 1, pp. 207–211, 2020. DOI: `10.1109/LCOMM.2019.2952602`.

[21] H. Park, S. Park, T. Song, and S. Pack, "An Incremental Multicast Grouping Scheme for mmWave Networks with Directional Antennas", *IEEE Communications Letters*, vol. 17, no. 3, pp. 616–619, 2013. DOI: `10.1109/LCOMM.2013.011513.122519`.

[22] D. Gong, Y. Yang, and H. Li, "Link-Layer Multicast in Smart Antenna Based 802.11n Wireless LANs", in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, 2013, pp. 484–492. DOI: `10.1109/MASS.2013.28`.

[23] Y. Niu, L. Yu, Y. Li, Z. Zhong, and B. Ai, “Device-to-Device Communications Enabled Multicast Scheduling for mmWave Small Cells Using Multi-Level Codebooks”, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2724–2738, 2019. DOI: `10.1109/TVT.2018.2883641`.

[24] A. Taya, T. Nishio, M. Morikura, and K. Yamamoto, “Concurrent Transmission Scheduling for Perceptual Data Sharing in mmWave Vehicular Networks”, *CoRR*, vol. abs/1903.09402, 2019. arXiv: `1903.09402`. [Online]. Available: `http://arxiv.org/abs/1903.09402`.

[25] G. Hong, Sim, M. Mousavi, L. Wang, A. Klein, and M. Hollick, *Joint Relaying and Spatial Sharing Multicast Scheduling for mmWave Networks*, 2019. arXiv: `1907.13085 [cs.NI]`.

[26] Y. Chang, Q. Liu, X. Jia, and K. Zhou, “Routing and Transmission Scheduling for Minimizing Broadcast Delay in Multirate Wireless Mesh Networks Using Directional Antennas”, *Wirel. Commun. Mob. Comput.*, vol. 15, no. 1, pp. 87–99, Jan. 2015, ISSN: 1530-8669. DOI: `10.1002/wcm.2318`. [Online]. Available: `https://doi-org.ezproxy2.utwente.nl/10.1002/wcm.2318`.

[27] A. Yamamoto, K. Ogawa, T. Horimatsu, A. Kato, and M. Fujise, “Path-Loss Prediction Models for Intervehicle Communication at 60 GHz”, *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 65–78, 2008. DOI: `10.1109/TVT.2007.901890`.

[28] I. Toyoda and T. Seki, *Antenna Model and Its Application to System Design in the Millimeter-wave Wireless Personal Area Networks Standard*, 2011.

[29] IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), “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”, Standard, 2016, pp. 1–3534. DOI: `10.1109/IEEESTD.2016.7786995`.

[30] *Gurobi Optimizer Reference Manual*, Gurobi Optimization LLC, 2021. [Online]. Available: `http://www.gurobi.com`.

[31] M. Jones, D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and O. Suchý, “Parameterized Complexity of Directed Steiner Tree on Sparse Graphs”, *CoRR*, vol. abs/1210.0260, 2012. arXiv: `1210.0260`. [Online]. Available: `http://arxiv.org/abs/1210.0260`.

[32] M. Brazil and M. Zachariasen, *Optimal interconnection trees in the plane: theory, algorithms and applications*, English, ser. Algorithms and Combinatorics. Springer, 2015, ISBN: 978-3-319-13914-2.

[33] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, “Multicast routing for multimedia communication”, *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 286–292, 1993. DOI: `10.1109/90.234851`.

[34] *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data*, U.S. Department of Transportation Federal Highway Administration, ITS DataHub, 2016. DOI: `http://doi.org/10.21949/1504477`.

[35] T. Clausen and P. Jacquet, *RFC3626: Optimized Link State Routing Protocol (OLSR)*, USA, 2003.

[36] S. Voss, “Worst-case performance of some heuristics for Steiner’s problem in directed graphs”, *Information Processing Letters*, vol. 48, no. 2, pp. 99–105, 1993, ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(93)90185-C. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002001909390185C.

[37] N. Skorin-Kapov and M. Kos, “The application of Steiner trees to delay constrained multicast routing: a tabu search approach”, in *Proceedings of the 7th International Conference on Telecommunications, 2003. ConTEL 2003.*, vol. 2, 2003, 443–448 vol.2. DOI: 10.1109/CONTEL.2003.176945.

[38] N. Skorin-Kapov and M. Kos, “A GRASP Heuristic for the Delay-Constrained Multicast Routing Problem”, *Telecommunication Systems*, vol. 32, pp. 55–69, May 2006. DOI: 10.1007/s11235-006-8202-2.